Pontifícia Universidade Católica
do Rio de Janeiro

João Marcos Dusi Vilela

An Efficient Algorithm for the Adjacent Quadratic Shortest Path Problem with Application to Smooth Transmission Line Routing

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção.

Advisor : Prof. Bruno Fanzeres dos Santos
Co-advisor: Prof. Rafael Martinelli Pinto

Rio de Janeiro
April 2021

João Marcos Dusi Vilela

An Efficient Algorithm for the Adjacent Quadratic Shortest Path Problem with Application to Smooth Transmission Line Routing

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção. Approved by the Examination Committee:

Prof. Bruno Fanzeres dos Santos
Advisor
Departamento de Engenharia Industrial – PUC-Rio

Prof. Rafael Martinelli Pinto
Co-advisor
Departamento de Engenharia Industrial – PUC-Rio

Dr. Sergio Granville
PSR – Soluções e Consultoria Ltda

Dr. Claudio Contardo
IBM – Toronto Software Lab

Rio de Janeiro, April 16th, 2021

João Marcos Dusi Vilela

Graduated in Industrial Engineering by Pontifícia Universidade Católica do Rio de Janeiro with one-year experience as an exchange student at the University of Southern California. Had a one-year experience as an intern at the Climate Policy Initiative (CPI - Rio de Janeiro) and worked as a research assistent at USC for three months. Also, had a final internship experience at PSR, in which is currently working as analyst and software developer.

Bibliographic data

Vilela, João Marcos Dusi

An Efficient Algorithm for the Adjacent Quadratic Shortest Path Problem with Application to Smooth Transmission Line Routing / João Marcos Dusi Vilela; advisor: Bruno Fanzeres dos Santos; co-advisor: Rafael Martinelli Pinto. – 2021.

v., 65 f. : il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2021.

Inclui bibliografia

1. Engenharia de Produção – Teses. 2. Problema de Caminho mais Curto;. 3. Custos Quadráticos;. 4. Teoria dos Grafos;. 5. Roteamento de Linhas de Transmissão;. 6. Sistemas de Informação Geográfica.. I. Santos, Bruno Fanzeres dos. II. Martinelli Pinto, Rafael. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. IV. Título.

CDD: 658.5

Acknowledgments

I am grateful to,

the Creator of all things, for giving me the joy of exploring such a complex and incredible world,

my amazing wife, Luisa, for choosing to stay by my side every day and for supporting me on all the trying times,

my parents, Wilson and Rossana, for loving me since the beginning and for always being there for me,

my siblings, Esther and Tiago, for sharing life with me and inspiring me to be a better brother,

my advisors, Bruno and Rafael, for leading me through the challenging paths of research and for contributing to this work,

my friends, who supported and pushed me to achieve greater things,

my friends and colleagues at PSR, who taught the value of knowing and understating how the world works,

my cat, Sinai, for making me company throughout the nights of research.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Vilela, João Marcos Dusi; Santos, Bruno Fanzeres dos (Advisor); Martinelli Pinto, Rafael (Co-Advisor). **An Efficient Algorithm for the Adjacent Quadratic Shortest Path Problem with Application to Smooth Transmission Line Routing**. Rio de Janeiro, 2021. 65p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

This dissertation explores the problem of transmission line (TL) routing through finding the shortest path on an undirected graph with no improving cycles, considering quadratic costs for adjacent arcs. This problem is known as the Adjacent Quadratic Shortest Path Problem (AQSPP). This work provides the theoretical background for the AQSPP, proposes an extension of Dijkstra's algorithm (aqDijkstra) for solving AQSPP in polynomial-time and discusses how AQSPP can be included in routing methodologies. Furthermore, it is presented an improvement to the algorithm: the adjacent quadratic A* (aq A*) with a backward search for cost-to-go estimation, to speed up search. For computational experiments, aqDijkstra and aqA* are benchmarked with other algorithms from the technical literature. The search behavior of the algorithms is also studied within different tests, including: quadratic cost variation, randomly generated graph instances and increasingly larger instances. The numerical results suggests that: (i) aqA*outperformed all the other algorithms, being 40 times faster than aqDijsktra and 50 times faster than the fastest benchmark algorithm; (ii) the studied algorithms do not lose efficiency as quadratic costs increase; (iii) aqA* and aqDijkstra were faster benchmark algorithms under random graph instances, indicating their robustness. Two applications are provided, one for illustrative purposes, and another to study performance on a real application. The aqA^* algorithm solved an AQSSP on a graph with almost a billion quadratic arcs and provided a route with three times lower additional costs.

Keywords

Shortest Path Problem; Quadratic Costs; Graph Theory; Transmission Line Routing; Geographic Information Systems.

Resumo

Vilela, João Marcos Dusi; Santos, Bruno Fanzeres dos; Martinelli Pinto, Rafael. **Um Algoritmo Eficiente para o Problema de Caminho mais Curto Quadrático Adjacente com Aplicação no Desenho de Rotas Suaves de Linhas de Transmissão**. Rio de Janeiro, 2021. 65p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Essa dissertação explora o problema roteamento de linhas de transmissão (LT) através da solução do caminho mais curto em um grafo sem ciclos de melhoria, considerando custos quadráticos para arcos adjacentes. Esse problema é conhecido como o Problema do Caminho Mínimo Quadrático Adjacente (CMQA). Esse trabalho apresenta uma descrição teórica do CMQA, propõe uma extensão do algoritmo Dijkstra (aqDijkstra) para solução de CMQA em tempo polinomial e discute como o algoritimo pode ser utilizado em metodologias de roteamento de LT. Em seguida, apresentamos uma melhoria estendendo o algoritmo A* para sua forma adjacente quadrática (aqA*), incluindo uma etapa de busca reversa para estimação de custos de chegada. Foram feitos experimentos computacionais contemplando a variação de custos quadráticos, geração de instâncias aleatórias, testes de estresse e comparação com abordagens já utilizadas na literatura. Os resultados sugerem que: (i) aqA* teve o melhor desempenho, atingindo tempos de busca 40 vezes mais rápidos que aqDijkstra e 50 vezes mais rápido que a abordagem mais rápida apresentada pela literatura; (ii) a eficiência dos algoritmos não foi afetada pela variação dos custos quadráticos; (iii) os algoritmos propostos aqA* e aqDijkstra também foram mais eficientes nas instancias aleatórias, reafirmando a superioridade dos mesmos. Duas aplicações são apresentadas, uma de objetivo ilustrativo e outra para um caso real. O algoritimo aqA^* foi usado para solução de um CMQA em um grafo de quase um bilhão de arcos quadraticos, resultado em uma rota proposta com custos adicionais três vezes menor.

Palavras-chave

Problema de Caminho mais Curto; Custos Quadráticos; Teoria dos Grafos; Roteamento de Linhas de Transmissão; Sistemas de Informação Geográfica.

Table of contents

List of figures

List of tables

List of Abreviations

i – Index of nodes
s – Index of source nodes
t – Index of target nodes
(i, j) – Indices of arcs
G – Graph
V – Set of nodes
A – Set of arcs
$|V|$ – Number of nodes
$|A|$ – Number of arcs
G' – Auxiliary Graph
V' – Set of auxiliary nodes
A' – Set of auxiliary arcs
c_{ij} – Cost of arc (i,j)
q_{ijk} – Quadratic cost of adjacent arcs (i, j) and (j, k)
θ_{ijk} – Deflection angle provoked by adjacent arcs (i, j) and (j, k)
$\Gamma(\theta)$ – Penalty function for deflection angles
$\delta(i)^+$ – Function to map successors of node i
$\delta(i)^-$ – Function to map predecessors of node i
x – Vector of binary decision of choosing arc (i,j) for the optimal path
c – Vector of costs for choosing arc (i,j) for the optimal path
b – Vector of right-hand-side values for the nodal flow-balance constraints
Q – Matrix of quadratic costs for selecting arcs (i,j) and (k,l) for the optimal path

1
Introduction

1.1
Motivation

A recurrent challenge in most power systems around the globe is to plan the expansion of its electrical transmission network, seeking for an appropriate accommodation of the constantly growing demand level and variability in energy production, within the daily system operation. Furthermore, the rapid development of renewable technology has been increasing the complexity of planning studies and demanding fast development of new transmission line projects.

The problem of defining when and what type of transmission facilities to build in terms of minimizing costs and maximizing social and net economic benefits has been a challenge for the power industry (Hobbs et al., 2016). Timely and cost-effective transmission expansion is required to provide secure and reliable electricity service to customers, improving competition, and ensuring efficiency in electricity markets (Velasquez et al., 2016).

According to Lumbreras et al. (2017), the optimal selection of the electrical elements to be installed in order to meet the objectives of the system as efficiently as possible is a fundamental issue on the transmission planning process and has been receiving significant attention in the past decade . Specially since one of the key drivers for transmission expansion is the integration of new renewable generation, which is commonly far from the load center, thus demanding increasingly larger transmission lines to flow energy.

As part of this expansion plan, different candidates for transmission lines (TL) are studied, and only the ones which meet the system's operation in the long run at minimum cost of investment are selected to be built. To achieve a consistent expansion plan, both in technical and economic terms, it is critically important to the system's planner a proper description of the candidate transmission line route and its respective construction costs (Santos et al., 2019). A better definition of the route can provide more realistic estimates towards the real cost of investment to be considered when expanding the system (Gonçalvez et al., 2021).

In fact, an accurate route evaluation of new power lines is a very complex task, which has been fundamentally tackled by mixing expert knowledge with limited spatial dataset. However, the current manual-based approach may not adequately assess the implementation and construction costs of the endeavor due to its intrinsic large-scale characteristics. More precisely, from a technical perspective, the impact of multiple topological aspects (e.g., nearby hydrology and hilly terrain) in the process of identifying the optimal transmission route and implementation costs must be carefully evaluated. For instance, the varied terrain composition and potential richness of water bodies nearby the route significantly increase the degree of construction complexity. Additionally, social and environmental factors induce legal obstacles to the route design. Areas with environmental preservation, urban and traditional communities, for example, are protected by the government and should be avoided.

It is often necessary to identify alternative detours on the tracing to avoid overrunning the administrative boundaries of these areas, which consequently affects the project's budget. As a result, routes commonly identified by planners and their respective construction costs are frequently far apart from the ones that will be implemented, which may delay or, in extreme cases, suspend the project. We also highlight that this planning inconsistency is worsened in the current context of most power systems since renewable sources have been mostly placed in locations far from the bulk system, thus necessitating of new and costly-efficient transmission infrastructure.

The process of defining a route begins with a three-dimensional topological map (digital satellite image) and a set of spatial and technical construction constraints, which are used to estimate a cost map. A weighted graph is built by assigning each pixel of the image to a node and connecting them to their node-neighbors with a weighted edge. The origin and target nodes represent physical location of the substations to be connected by the TL. Edge weights are defined as the cost of building part of the transmission line between neighbor nodes. This cost varies depending on geographical, technical, social, environmental, and economic aspects (Monteiro et al., 2005).

The optimal route can be found through calculating the least-cost path between origin and target nodes. This problem can be classified as a Shortest Path Problem (SPP), which is deeply explored by literature. Many authors have been proposing different approaches to solve this problem. The most popular solution comes from Dijkstra's work (Dijkstra, 1959), where a dynamic programming algorithm that guarantees global optimality in polynomial time was proposed.

Using SPP to design TL routes has been, so far, the state-of-the-art

approach in this context. However, SPP presents limitations toward some modeling technical constraints, such as the requirement of avoiding sharp curves on the route. The high deflection angles presented by SPP route solution are not realistic, since the towers require to support these angles are expensive (Kiessling et al., 2004; Fang et al., 1999) and only set if necessary. These constraints are deeply important for TL design but are usually ignored in first stages of planning (Piveteau, 2017).

To properly model this technical and economical constraint, the work discusses how TL routing can be formulated as a special case of the Quadratic Shortest Path Problem (QSPP) called Adjacency Quadratic Shortest Path Problem (AQSPP), which only considers the interaction cost between adjacent edges. An efficient algorithm called Adjacent Quadratic Dijkstra ($aqDijkstra$) was developed to is solve AQSSP. Two improvements are also proposed, aiming to enhance speed and memory-management. There are known approaches which could be used to solve AQSPP as well, but only for small or medium graph instances. Since TL routing problems are intrinsically very large (e.g. 10^6 nodes and 10^7 arcs), larger problems start to become intractable. Overcoming this obstacle was one of the key motivators for this research.

Through AQSPP formulation, penalties for deflection angles can be modeled as costs of quadratic arcs, proxying real building practices. Since higher deflection angles imply stronger and more expensive tower structures, engineers usually avoid sharp curves on proposed routes. Although there are recent approaches in the literature (Piveteau, 2017; Santos et al., 2019; Gonçalvez et al., 2021) that attempt to avoid sharp curves on the routing process, no work was found to use AQSPP formulation to explicitly model penalties for deflection as quadratic costs.

1.2 Objectives and Contributions

This Master's dissertation introduces a novel approach for transmission line routing which considers curvature penalties according to deflection angles supported by towers along the route. To accomplish that, the routing problem is formulated as an AQSPP, modeling the penalties as quadratic arc costs.

The contributions of this work to the scientific community can be listed as:

- Development of two efficient algorithms to solve AQSPP.
- Novel methodology for transmission line route optimization, formulating the problem as an AQSPP with route curvature penalties.

– Implementation of an open-source Julia package containing the proposed algorithms and other benchmarks.

1.3 Thesis Organization

This thesis will be structured as follows:

– Chapter 2 provides a theoretical background to routing optimization algorithms.

– Chapter 3 describes the methodology used for transmission line routing and presents the proposed algorithms.

– Chapter 4 discusses the results from four computational experiments designed to study the proposed algorithms.

– Chapter 5 presents an illustrative and real application cases of transmission line routing in which the proposed algorithms are used.

– Chapter 6 concludes this thesis by summarizing the main findings and exploring possible subjects for future research.

2
Theoretical Background

Optimizing transmission line routes is a common spatial routing application. The main idea is to find a physical route between two locations, usually associated with spatial coordinates, considering geographical data, such as terrain, forests, rivers, and other relevant attributes. A more detailed routing also considers technical, economic and social aspects. Once this information is collected and processed, several techniques can be applied to optimize the route. These techniques are usually implemented on Geographic Information System (GIS) computational products. Section 2.1 introduces the problem of finding the optimal route of transmission line through a review of the literature. Then, Section 2.2 focus on the Shortest Path Problem (SPP) as the most common approach to formulate routing problems. A novel formulation of the problem as an Adjacent Quadratic Shortest Path Problem (AQSPP) is presented on Section 2.3 to tackle the problems presented on Section 2.2. Furthermore, Section 2.4 formalizes the problem.

2.1
Transmission Line Routing

Routing new electrical transmission lines is a complex activity. When done manually, it is a time consuming and costly design that requires massive and very detailed spatial information and project engineers experienced in the terrain. Usually, engineers who perform TL routing aims to optimize the route by minimizing construction costs subject to geographic, environmental, social, and legal constraints. Thus, routing can be defined as the previous stage to the design of a new electric power line, where the planner decides the path and areas crossed by the line considering contextual spatial constraints (Monteiro et al., 2005).

The automation of the routing process was the natural path towards improving new TL designs, as spatial data became more detailed and GIS use spread out in the scientific community. In this context, technical literature on distribution and transmission systems' planning benefited from these improvements, as discussed by Choobineh and Burgman (1984). Among the benefits, smaller efforts in the revision of the TL project and capabilities for studying

multiple routing solutions should be highlighted.

In the first approaches to TL routing, K-shortest paths (KSP) and goal programming (GP) were explored by (Choobineh and Burgman, 1984). KSP uses a brute-force method of systematically listing all possible routes between origin and destination nodes, guaranteeing all possible paths are found within an accumulated cost threshold (Shandiz et al., 2018). Although global optimality is achieved, the algorithm is not practical for medium or large problems in computational performance terms, as run-time and memory requirements increase factorially.

Further on, approaches based on graph theory for least-cost path finding became common in the research community, such as Iterative Penalty Method (Akgün et al., 2000) and Dijkstra (1959). The overall decision-making process is best described by an evaluation of possible paths from an origin point (or substation) to a destination point, considering costs of passage for TL throughout the paths.

Researchers have been exploring other approaches, such as Analytic Hierarchy Process (AHP) and Fuzzy AHP (Eroglu and Aydin, 2015), multi-criteria with Qlearning (Demircan et. al., 2011), and, improved genetic and artificial bee colony algorithms (Eroglu and Aydin, 2018). The motivation for this work is related with technical problems that current routing approaches are not able to handle, which is to avoid unrealistic curves throughout the optimization.

In addition, claims from Eroglu and Aydin (2015) and Demircan et. al. (2011) relies on the complexity of calculating realistic cost estimates for constraints, arguing that some constraints cannot be associated with monetary values, such as areas where traditional communities live. AHP and multi-criteria are multiple decision techniques for determining the criteria to be taken into consideration when performing the routing process. Although it can be a powerful tool for analysis and candidate creation, especially when comparing routing alternatives, it is still deeply dependent on a wide variety of detailed spatial data and expert knowledge.

Most approaches are based on the same idea, finding the TL spatial route that minimizes costs or maximizes benefits. The costs and benefits are usually associated with monetary values (Monteiro et al., 2005) or relevancy indexes (Eroglu and Aydin, 2015), depending on the approach and financial data available.

2.2 Route Optimization as a Shortest Path Problem

Finding the shortest path in a graph is a well-known problem and has been deeply explored by the literature (Madkour et al., 2017). This problem is usually referred as the Shortest Path Problem (SPP) and aims to find a path P on a graph G between a source node s and target node t that minimizes total cost of the path. This problem has a linear objective function of $c^T x$, where c is a cost vector and x is a binary vector of arc selections.

The Shortest Path Problem can be formulated as

$$\min_{x \in \chi} c^T x \tag{2-1}$$

where χ is the set of feasible binary vectors

$$\chi := \{x \in \mathbb{R}^m | Bx = b, x \in \{0,1\}^m\} \tag{2-2}$$

The matrix B is a $|V| \times |V|$ matrix which maps the connection between the nodes of the graph. The array b sets 1 to the source node, -1 to the target node and 0 the other nodes.

Although many approaches have been developed to solve SPP, such as Bellman-Ford (Bellman, 1958), Floyd-Warshall (Floyd, 1962) and A^* (Hart et al., 1968) algorithms, the most well-known approach is due to Dijkstra's work (Dijkstra, 1959). The Dijkstra's algorithm solves the shortest path problem from a given node to all other nodes in a graph. The algorithm assumes non-negative weights and uses a labelling structure on the search process. On each iteration, the node being evaluated can receive two types of labels: visited or not visited. It initially sets the source node as visited and checks their neighbor nodes for the not visited label. Then, the shortest arc is identified, and its corresponding node is labeled visited. The algorithm iterates until all nodes receive the label visited. It should be said that stopping criteria can be added to the algorithm to interrupt the search after visiting a target node. The Dijkstra's algorithm has a time complexity of $O(|V|^2)$. The algorithm of Dijkstra's is presented on Algorithm 1.

2.3 A Novel Approach: The Adjacent Quadratic Shortest Path Problem

The linear nature of SPP prevents it to tackle more complex problems, such as establishing network protocols (Murakami and Kim, 1997) and optimizing vehicle routing (Martinelli and Contardo, 2015), that have a quadratic order. The Quadratic Shortest Path Problem (QSPP) extends SPP by finding a path between s and t that minimize sum of costs of arcs and the sum of

Algorithm 1: Dijkstra's Algorithm

input : G, c, s, t
output: D, P

```
initialize D_i;
initialize Q_i;
initialize P_i;
D_s ← 0;
for each node v in V do
    if v ≠ s then
        D_v ← Inf;
while Q_i ≠ ∅ do
    (i) ← argmin_i{Q};
    remove (i) from Q_i;
    if i == t then
        stop while;
    for each j in neighbors of i do
        C_j ← D_i + c_ij;
        if C_j < D_j then
            D_j ← C_j;
            Q_j ← C_j;
            P_j ← i;
```

interaction costs over distinct pairs of arcs on the path. Thus, the problem has a quadratic objective function that can be defined as $x^T Qx + c^T x$, where Q is a quadratic matrix with costs of pair arcs.

In this work, a special case of QSPP is studied, the Adjacent Quadratic Shortest Path Problem (AQSPP), which has the same quadratic objective function, but Q only considers the interaction cost between adjacent arcs. This change introduces great sparsity on Q and makes the problem significantly smaller. The original QSPP is a NP-hard optimization problem that seeks to find the path that minimizes the sum of arc's costs and arc's pairs interaction costs, over a graph (Rostami et al., 2015).

The theory and applicability of QSPP have been studied through the works of Rostami et. al. (2015 and 2018), Hu and Sotirov (2017 and 2018) and Buchheim and Traversi (2015). Buchheim and Traversi (2015) explores a more general solution for binary quadratic problems (BQP) and proposes an exact algorithm to solve special a case, based on branch-and-bound and lower bounds estimates. Rostami provides in-depth mathematical description of QSPP and proposes an algorithm to solve AQSPP in polynomial-time (Rostami et al., 2015), which is based on a transformation that reduces AQSPP to SPP by

modifying the original graph.

Furthermore, the authors Hu and Sotirov (Hu and Sotirov, 2018) also characterize QSPP mathematically and derive an algorithm that examines whether QSPP instances of directed grid graph are linearizable. In other words, the authors study when a grid graph with quadratic cost can be transformed into graph with linear costs. They also reviewed Rostami's work and prove that, despite stated on (Rostami et al., 2015), their algorithm does not find optimal solutions for direct graphs in polynomial time, but only for direct acyclic graphs.

In this context, this dissertation study AQSPP for a more generic type of graph, which cannot have improvement cycles. Improvement cycles are cycles that can lower total cost once they are found by the path search algorithm and due to the quadratic cost structures, they are non-negative cost cycles. Graphs with no improvement cycles can be considered a more general type of graph, since directed acyclic graphs do not have cycles, but the opposite does not hold. These types of graphs can be found in several contexts and applications. In particular, graphs that represent spatial attributes (terrain, distance, slope etc) are widely used on route-planning applications and do not have improvement cycles.

The use of AQSPP to model transmission line routing problems allows a better representation of penalties for TL route curves through adjacent quadratic functions of AQSPP. Although there are approaches in the literature that attempt to avoid sharp curves on the routing process, no work that used AQSPP formulation to explicitly model penalties for deflection as quadratic costs was found.

2.4 The Adjacent Quadratic Shortest Path Problem

Let a weighted graph be defined by $G = (V, A)$, with V as the set of nodes i and A as the set of arcs (i, j). We assume a node s as the source node and t as the target node, such that a $s - t$ path $P = \{i, j, ..., k\}$ is a order set of nodes from $s = i$ to $t = k$. We define a linear function $c : A \to R^+$, which maps arcs into a cost and a quadratic function $q : A \times A \to R^+$, which maps every pair of arcs into a cost. Both linear and quadratic functions can only return strictly positive costs. For the AQSPP, we assume that the quadratic costs of all non-adjacent pair of arcs are zero.

Furthermore, the predecessors and successors of each node i are found through the functions $\delta(i)^- = \{j \in V|(j, i) \in A\}$ and $\delta(i)^+ = \{j \in V|(i, j) \in A\}$, respectively. Finally, a binary variable $x_{ij} \in \{0, 1\}$ is set to indicate the

presence of arc (i, j) on the optimal path. When an arc is selected for optimal path, the associate linear cost is incorporated on the objective function (2-3). The same goes for the selection of adjacent arcs (i, j) and (k, l) given that $j = k$, which adds a quadratic cost q_{ijkl} to the objective function.

Therefore, the quadratic formulation is as follows:

$$Min \sum_{(i,j)\in A} \sum_{(k,l)\in A} q_{ijkl} x_{ij} x_{kl} + \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{2-3}$$

s.t.

$$\sum_{j\in\delta(i)^+} x_{ij} - \sum_{j\in\delta(i)^-} x_{ji} = b_i \quad \forall i \in V \tag{2-4}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \tag{2-5}$$

with $b_i \in \{-1, 0, 1\}$, where $b_s = 1$, $b_t = -1$ and $b_i = 0$ otherwise.

This problem results on a path that connects nodes s and t at minimum cost. The optimal path is an ordered set of nodes $P^* = \{s, i, j, ..., k, t\}$ such that $i \in V$, $(i, j) \in A$ and $s \neq t$. The constraint (2-4) establishes the connection between nodes through balance equations, meaning that every unit arriving or placed at node i must leave or be consumed. Thus, setting $b_s = 1$ and $b_t = -1$ forces the arcs x_{ij} of a feasible path P to be equal to one, so that constraints (2-4) are respected, and the unit placed on b_s arrives at b_t through P.

It should be noticed that the feasible region defined by (2-4) and (2-5) on AQSPP formulation is the same from SPP. Therefore, all the feasible paths considered on SPP are also feasible on ASQPP. The main difference in formulations come from introducing the quadratic term $q_{ijkl}\ x_{ij}\ x_{kl}$ on the objective function. Once a feasible path is evaluated, the costs of all pairs of adjacent arcs on the path are considered.

The problem can also be studied through matrix notation, such that the objective function becomes $x^T Q x + c^T x$, where Q is a matrix $|A| \times |A|$ that represents the interaction costs between pairs of arcs, x is a binary vector for the selection of arcs and c a vector for their linear costs. This matrix notation is commonly used on the quadratic programming (QP) literature since it allows the study of the whole problem by looking only at matrix Q individually. On AQSSP, for example, we can assume Q to be symmetric, positive semi-definite and very sparse.

Using QP techniques to solve AQSPP is a valid approach and have been explored in literature (Buchheim and Traversi, 2015; Caprara, 2008). However, we argue that as the size of graph instances grows, using traditional QP solvers becomes computationally challenging or impracticable. Memory issues to store matrix Q, the constraints defined by (2-4) and temporary data required by the solver can easily occur. These limitations motivated the study of this problem

with an algorithmic perspective, ultimately resulting in the development of two algorithms to solve AQSPP. The algorithms are described in Section 3.2.1 and Section 3.2.2.

2.4.1 Linearization Technique

In order to solve the AQSPP, Rostami et al. (2015) proposes a polynomial-time algorithm based on the reduction of AQSPP on graph G with (V, A, Q) to a SPP on a linearized graph G' with (V', A'). The reduction is as follows.

Let the nodes and arcs from G' be

$$V' = \{(s,s)\} \cup \{(i,j) : i,j \in A\} \cup \{(t,t)\} \tag{2-6}$$

$$A' = \{((i,j),(j,k)) : (i,j),(j,k) \in V'\} \tag{2-7}$$

Arcs are associated to transformed nodes $((i,j),(j,k)) \in A'$, so that new arc weights are defined as:

$$w(i,j,k) = \begin{cases} c_{jk} + q_{ijk} & (i,j) \neq (s,s) \wedge (j,k) \neq (t,t) \\ c_{jk} & (i,j) = (s,s) \\ 0 & (j,k) = (t,t) \end{cases} \tag{2-8}$$

The Figure 2.1 illustrates the transformation of graph G (a) to graph G' (b). The basic idea is to convert original arcs (i,j) into nodes n_{ij} and map quadratic costs to the connection of the new nodes. Notice that the source and target nodes are redefined as pairs to themselves i.e. (s,s) and (t,t), with new arcs arriving to (t,t) at zero cost and leaving (s,s) at c_{sj} cost.

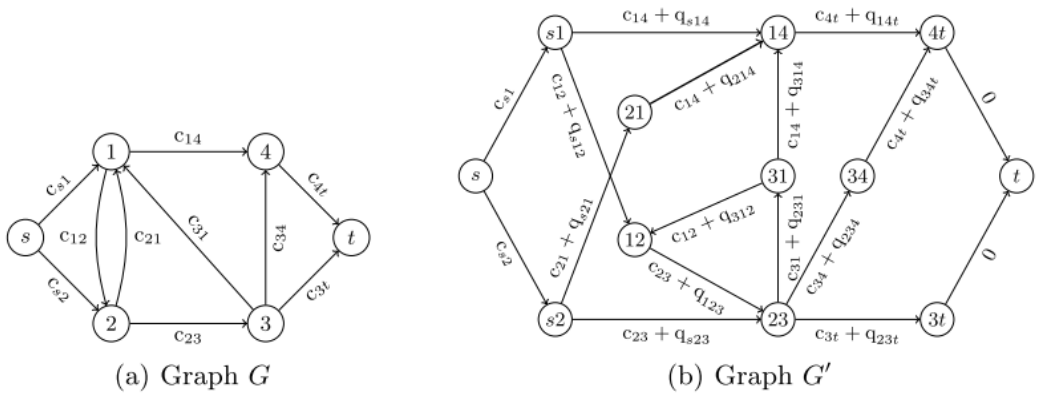


Figure 2.1: Linearization of Graph G into graph G' (Rostami et al., 2015)

Once the graph is linearized, the quadratic problem can be reduced to a linear problem. In other words, by transforming G, the AQSPP is reduced to a SPP, thus solvable by any algorithms that solves SPP, such as Dijkstra's Algorithm. The authors claim that this approach guarantees that G can be solved in $O(min\{|A|^2, |V|^3 + |A|log|A|\})$ time.

This algorithm is a possible solution to solve AQSPP on small to medium graph instances with acceptable performance. However, the linearization technique does not scale efficiently since it requires an expansion of the original graph according to quadratic costs. For LT routing applications, graph instances are intrinsically huge, easily achieving dimensions of 10^6 nodes, 10^7 arcs and 10^8 quadratic arcs. Performing this technique on graph of such dimensions would not only be memory-intensive but eventually non tractable. With that in mind, the next chapter presents novel AQSPP algorithms, which are efficient and scalable.

3
Proposed Methodology

3.1
Spatial Data Setup

The solution integrates geographical data, geoprocessing techniques and dynamic programming to find the least-cost path between two substations. All spatial data processing was made using R's spatial data libraries, as well as other R resources and functionalities. Although it was not built primarily as a GIS software, it matches all requirements necessary to perform common and advanced GIS spatial processing.

As input data, terrain and social-environment spatial information was obtained from government institutions which are publicly available. A four-step process is proposed: selecting relevant spatial data for study, constructing a cost map, converting it into a weighted graph and finding the least-cost path between two nodes (coordinates).

3.1.1
Data Selection

The spatial data used can be described with two types of representation models: matrix (raster) and vector (shapefile). The first model, as the name suggests, organize the data as georeferenced matrices, in which area is divided as a rectangular grid of regular cells (pixels). Each cell is associated with a unique coordinate and a numeric value associated. These values can represent quantitative or qualitative data. On the other hand, vector model represents elements as points, lines or polygons. The position of each object is defined by its location in space, according to a coordinate reference system (CRS). Spatial objects do not fill the entire space, which means not all positions in space need to be referenced in the model.

When working with spatial analysis, a wide variety of geographical aspects can be analyzed, some related to topology, some concerning legal and administrative limits, and other types of relevant data. Therefore, it is important to select only those which are relevant to route definition. The list

of variables should be selected according to the region being studied. In other words, it is necessary create a spatial delimitation on the region.

Every construction process that extends over large territorial area faces project constraints. In transmission expansion planning, many TL routes must be remade or adjusted, so that technical limitations and legal barriers can be overcome. Projects such as roads and railways constructions, for example, involves construction along the whole territory and faces similar problems. These constraints can be divided in two groups, environmental and social constraints. Each have distinct influence over construction decision, which goes from route retracing to area contouring, and even project infeasibility.

In the environmental perspective, Conservation Units can be classified as spatial territories protected by the government, due to their limited or special natural resources. To overlook the area and guarantee its protection, every unit usually has a government agency responsible for its administration and supervision. Depending on how a transmission line route crosses the area, costs can vary and increase substantially, forcing the route to contour it. Costs related to authorization and construction specificities (higher towers for example), can make avoiding the area cheaper than trespassing the conservation unit.

Moreover, defining paths through water bodies requires special infrastructure and increase construction expanses, thus not preferable. Crossing wide water bodies, for example, require much higher towers than normal, which are very expensive and harder to setup. A famous case is the Brazilian TL Tucuruí-Macapá-Manaus, which crosses the Amazon River margin to margin, where conductor cables are supported by two towers, 2.5 kilometers from each other, and 300 meters high (taller than the Eiffel Tower).

On the social aspect, some communities described as traditional communities have their territory legally delimited and protected by the government. Interfering in these communities, such as indigenous territories, are crucial aspects in licensing process by environmental agencies.

Rural settlements also have great impact on routing. It generally refers to small rural properties and high habitational density in rural areas. Building power transmission lines in these areas have been shown to be a challenge for construction companies. Not only physical risks for residents should be taken into consideration, but line's right-of-way can interfere with production activities and have negative impact on income generation. When defining the route, urban areas must be avoided at maximum. Building transmission lines close to habitational concentration is challenging, expensive and offers risk to residents. Urban growth is also a factor to be considered since an unpredicted urbanized region through original tracing can implicate in redesigning the line's

tracing and increasing project's expenses.

3.1.2 Building a Cost Map

After selecting and setting up the required spatial data for analysis, a map of costs (or weights) should be constructed. The basic structure of a cost map is defined by a numerical matrix, where every element of the matrix is associated with a coordinate and a cost value. The definition of costs depends on available data and initial assumptions. This methodology assumes the existence of a reference construction cost per kilometer, which will be used further on to estimate an initial cost map.

Usually, when spatial analysis requires topological information, such as height or slope values, a mix of matrix and vector models is commonly used, and also recommended. Topological data is set as a base map and relevant spatial areas are layered on top, delimiting areas of interest. Overlaying and clipping techniques are widely used to subset and reduce maps (matrix model) considering different spatial limits (vector model). For example, if a base map includes data a whole country but the studied area only concerns a certain state, the area that defines geographically the state can be used to subset (clip) the base map.

This methodology proposes a two-step process for cost map construction: building a base cost map and including additional costs. The first concerns building an initial cost map based on terrain slope and reference construction cost. Terrain slope can be described as the terrain's inclination of a given area. In other words, the slope measures the altitude variation of a region. Figure 3.1 illustrates a map of slope values, where each value represents the average slope of the area within an image pixel. Assuming cells of equal dimensions, the linear estimation of cell length, considering terrain slope, can be calculated. Finally, given the reference cost and linear length approximations, the base cost map is calculated, as illustrated on Figure 3.2.

Figure 3.1: Terrain Slope Map (3D)

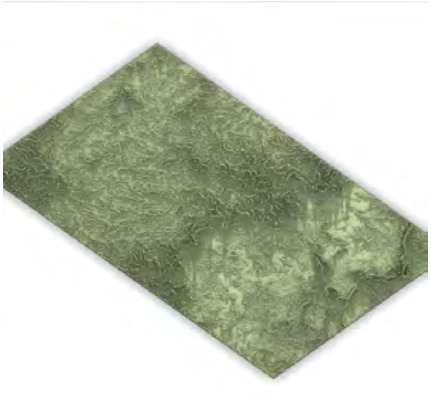


Figure 3.2: Base Cost Map (3D)

Although slope values are intuitive, it is not common for routing methodologies to be based on slope estimations. However, it is a fact that terrain slope is intrinsically related to geographic distance. The proposed methodology uses this concept to convert slope into linear distance, which will further be used to estimate construction costs.

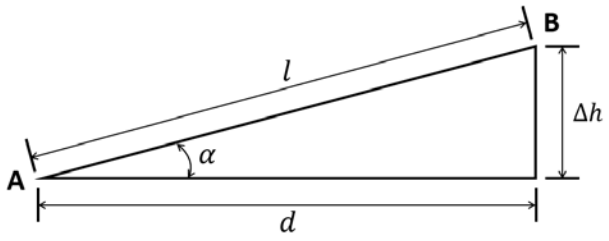


Figure 3.3: Linear approximation of terrain slope

From Figure 3.3, the reference cost of a cell would be the product of the ramp distance I $[m]$ and the construction cost c $[\$/m]$.

$$I_{ij} = \sqrt{\Delta H_{ij}^2 + d_{ij}^2} \tag{3-1}$$

$$c_{ij}^{base} = I_{ij} C \tag{3-2}$$

Then, the studied region must be identified and delimited. Coordinates of both substations are obtained and placed on the map for visual verification. After that, a 10-kilometer buffer between substations a created in a form of vector data. Finally, all cells inside the delimited area are selected, creating a new map containing only the area inside the 10-kilometer buffer. This step is called raster cropping, and it is crucial for increasing processing speed. By cropping the map, the number of valid pixels is much smaller, thus resulting in less nodes and arcs on the converted graph. With smaller instances, path-search speed increases significantly.

It is reasonable to assume that some studies may less focused on the economic and technical aspects, and may prefer routes that fully respect environmental and social constraints, and seeks to minimize construction effort. Depending on the planner goals, different cost assumptions can be made to achieve realistic candidates for network expansion.

In practice, there are two ways of considering these assumptions. The first one is making costs inside the spatial area nonexistent, which can be interpreted as giving an "inaccessible" status to a node. The other one is attributing massive costs to the area, therefore making the solving methods avoid the area at maximum. Both approaches can work but making a node inaccessible can be problematic if origin or destination nodes are inside constrained areas. With that in mind, the second alternative was chosen.

Furthermore, the spatial constraints are modeled as vector objects, as it is shown on the Figure 3.4 for the colored areas. These types of spatial data can be obtained from several different sources, both public and private. However, for the purpose of generalization, it is assumed that the required spatial data is publicly available and can be obtained from government agencies. Spatial projections must be adjusted when necessary for a proper representation of regions within Brazil. Finally, additional costs associated with each constraint are added to the base cost map, resulting in the final cost map displayed on Figure 3.5.

Figure 3.4: Constraints Layers (3D)

Figure 3.5: Final Cost Map (3D)

3.1.3 Map Conversion

Once each map cell is properly associated with a cost, we can generate the required graph. Its structure, as described before, consists of a set of nodes, arcs (connection of two nodes) and weights. Figure 3.6 and Figure 3.7 present an visual representation of the conversion set. The map on Figure 3.6 is converted

into the graph of Figure 3.7, where the numbers inside the nodes represent the nodes indices and the black lines represent the arcs.

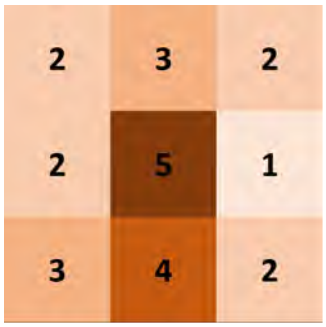


Figure 3.6: Map

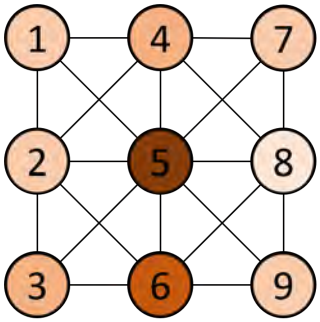


Figure 3.7: Graph

The transformation of a cost map in a weighted graph starts with representing each map cell as a graph node. Then, for every given pair of connected nodes (cell neighbors in a map) a graph arc must be created. In addition, the average costs between neighbors must be associated with the created arc. It is important to notice that diagonal arcs must have their weights adjust to correctly represent the distance between the nodes. The Figure 3.8 highlights this process. Notice that c_{14} is simply the average between nodal values of node 1 and 4. In contrast, the movement from node 1 to node 5 is diagonal, thus considering a $\sqrt{2}$ to the arcs cost c_{15}.

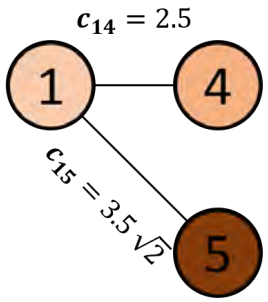


Figure 3.8: Final cost map

3.1.4 Curvature Penalties

In simple terms, transmission lines are electric towers connected by conductor cables. Depending on how and where the towers are placed spatially, high inflection angles can occur. Depending on the angle, stronger support structures must be added to the tower to handle the increased mechanical effort. These structures are often very expensive and can significantly impact

construction costs. The use of AQSPP brings innovation to LT routing, as it allows us to model penalties for deflection angles. Thus, we suggest an adjacent cost $q(\theta_{ijk})$ where θ_{ijk} is the inflection angle between two adjacent arcs (i, j) and (j, k).

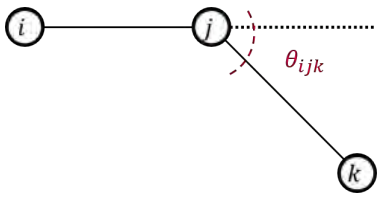


Figure 3.9: Tower Angles

For this work, the function is defined as

$$q_{ijk} = \Gamma(\theta_{ijk}) \tag{3-3}$$

Where

$$\theta_{ijk} = arccos(\frac{y_1 y_2}{\|y_1\| \, \|y_2\|}) \tag{3-4}$$

The deflection angles θ_{ijk} are associated with higher mechanical efforts, and consequently with more expensive structures. The most common towers and their respective angle range and relative costs are present on the Table 5.3.

3.2 Route Optimization

In this section, a proposed AQSPP algorithm named adjacent quadratic Dijkstra is described. The algorithm is an extension of Dijkstra algorithm to solve AQSPP. Many classic algorithms can be used for efficient path search, such as Dijkstra (Dijkstra, 1959) and A^* (Hart et al., 1968), but most of them have linear objective functions, thus not able to represent quadratic costs. We propose an extension of classical Dijkstra algorithm, which we call Adjacent Quadratic Dijkstra (aqDjikstra), that solves AQSPP in polynomial-time for graphs with no improvement cycles. The algorithm is based on the concept of node labels and can properly address adjacency costs.

3.2.1 Proposed Algorithm: Adjacent Quadratic Dijkstra

The Adjacent Quadratic Dijkstra algorithm follows a similar structure as the classic Dijkstra's algorithm, but modifies how labels and accumulated costs are defined. We denote a label as $l_j = \{(i, j)|i \in \delta(j)^-\}$, representing an arc that arrives at node j. Let D_{vl} be the accumulated cost on the pair node-label

(v, l) and $L(v)$ be a function that returns all labels associated with node v. This allows aqDijkstra to evaluate the shortest path to j that passes through (i, j). Therefore, when evaluating a connection between j and k, through label (i, j), we can map adjacent quadratic cost q_{ijk}.

The algorithm is presented in Algorithm 2.

Algorithm 2: Adjacent Quadratic Dijkstra

input : G, c, q, s, t
output: D, P

initialize D_{ij};
initialize Q_{il};
$D_{sl} \leftarrow 0$;
for *each node* v *in* V **do**
    for *each label* l *in* $L(v)$ **do**
        if $v \neq s$ **then**
            $D_{vl} \leftarrow Inf$;
            $P_{vl} \leftarrow$ *origin node from label* l;
while $Q_{il} \neq \emptyset$ **do**
    $(j, l_1) \leftarrow argmin_{il}\{Q_i\}$;
    remove (j, l_1) *from* Q_{il};
    if $j = t$ **then**
        stop while;
    for *each* k *in neighbors of* j **do**
        $l_2 \leftarrow$ *label index of arc* (j, k);
        $C_{kl_2} \leftarrow D_{jl_1} + c_{jk} + q_{ijk}$;
        if $C_{kl_2} < D_{kl_2}$ **then**
            $D_{kl_2} \leftarrow C_{kl_2}$;
            $P_{kl_2} \leftarrow j$;
            $Q_{kl_2} \leftarrow C_{kl_2}$;

The iteration over the labels is controlled by a priority queue Q. Priority queues are abstract data types similar to stacks in which each element has a priority assigned to it. These queue structures are often used to speed up search, and also exist on the original Dijkstra algorithm. The element with highest priority is always positioned on top of the stack, forcing it to be the first to be removed. In this case, the labels with smaller accumulated costs should be prioritized, since they indicate shorter paths, and should be positioned on top of the stack.

At each iteration, the label with smallest cumulative cost l_1 is removed from Q and used as reference to explore the neighbor nodes. Each neighbor k is then evaluated, the cumulative quadratic cost C_{jl_2} is calculated and

$\{Q_{jl_2}, D_{jl_2}\}$ gets updated, when evaluated costs are lower than current costs. The algorithm stops if Q gets empty or all labels from target node $L(t)$ were evaluated.

It is claimed that aqDijsktra's can provide optimal solution if it holds that graph instance G does not have any improvement cycles. This is a more widespread claim, since Hu and Sotirov (Hu and Sotirov, 2018) argues that G must be directed and acyclic. Considering that acyclic digraphs do not have cycles by definition, they meet aqDijsktra's optimality conditions and can be solved.

However, if we consider a weighted graph U, improvement cycles can exist and, thus, must be checked. The first approach to verified the existence of improvement cycles is to perform a complete search on the graph. Although it would provide an exact answer, this approach is extremely inefficient.

Another procedure would be to look for improvement cycles during aqDijkstra iterations. Although it would be significantly faster than a complete graph search, adding this verification step would have a negative impact on performance, reducing search speed.

A final approach to address this issue would be to solve G and search for any cycles within the solution path. Since by hypothesis there are no arcs with non-positive cost, every cycle within the solution path must be an improvement cycle. Therefore, there is no optimality guarantee for the solution path. We consider this to be the best approach for instances with well-known structures, like grid graphs, for example.

3.2.2 Algorithm Improvement: Adjacent Quadratic A^*

When developing algorithms for optimization problems, in special methods for graph search, two major aspects of computation must be considered: memory and speed. There is a sweet spot between an algorithm that is fast but extremely memory consuming, and an algorithm that is slow and allocates memory efficiently. Two improvements for aqDijkstra are proposed, one that speed up search and the other that reduces memory dependency. First, it is studied a natural extension of aqDijkstra based on A^* algorithm (Hart et al., 1968) and included cost-to-go estimations, i.e. completion bounds. Then, it is proposed the placing quadratic cost calculation within aqDijkstra and remove the need to pre-calculate all quadratic costs.

To increase search speed, a bidirectional search alternative is explored. This scheme consists in solving the problem in two steps: (i) a preliminary backward search with SPP, starting form target node t, to find cost-to-go

estimates; (ii) a forward search with ASQPP, from source s to target t node, considering the cost-to-go estimates as lower bounds for the search. This approached is inspired by the work of Thomas and Calogiuri and Hewitt (2019), where the authors extend Hart et al. (1968) results for A^* algorithm to develop a bidirectional search to solve resource-constrained shortest path problems. The idea is built upon Hart et al. (1968) demonstration that A^* only terminates with an optimal path when the estimates of the cost-to-go are actual lower bounds.

For the case of a quadratic shortest path search, linear estimations of costs-to-go are actual lower bounds for the quadratic problem and attend optimality conditions. The fundamental idea is that these linear estimations can be calculated through a backward search from the target node t to all other nodes with an SPP approach.

The algorithm starts with a backward search with SPP, which is a regular search that starts on the target node t and considers only linear arc costs. Thus, we use Dijkstra's algorithm to perform a complete search (visiting all nodes) and store the lowest cost from t to all the other nodes on the graph. These costs will be considered the costs-to-go estimates on the following step and are lower bounds for the quadratic problem. It is important that the backward search goes through the entire graph, so that cost-to-go estimates are available on every node and the optimality of the path is guarantee.

On the second step, we perform a forward search with ASQPP from source node s to target node t. Here, we introduce the adjacent quadratic A^* (aqA^*), an extension of aqDijkstra that considers cost-to-go estimates to decide which node to expand next. This structure is the same described by (Hart et al., 1968) when developing A^* algorithm. By estimating the cost of arriving at t, we can greatly reduce the search tree and speed up the procedure. However, it is important to be aware that these estimates must be lower bounds to the real path cost in order to guarantee admissibility. In other words, if the cost-to-go from i to t is not lower or equal to the minimum cost of going from i to t, the solution of aqA^* may not be globally optimal.

Since the costs-to-go come from a backward search with SPP, only linear costs are considered, guaranteeing lower bounds for aqA^*. The lower bound value of each node is included on distance calculation, so that for each iteration Q_{jl_2} is updated according to the original cost C_{jl_2} and the cost-to-go B_k from node i to target node t. Therefore, the farther the evaluated node is from the target, the greater the cost-to-go will be, and, consequently, the lower the priority will be on the priority queue.

The algorithm is presented on Algorithm 3.

Algorithm 3: Adjacent Quadratic A^* with backward cost-to-go estimation

input : G, c, q, s, t
output: D, P

$B \leftarrow Dijkstra(G, t)$;
initialize D_{ij};
initialize Q_{il};
$D_{sl} \leftarrow 0$;
for *each node* v *in* V **do**
  for *each label* l *in* $L(v)$ **do**
    if $v \neq s$ **then**
      $D_{vl} \leftarrow Inf$;
      $P_{vl} \leftarrow origin\ node\ from\ label\ l$;
while $Q_{il} \neq \emptyset$ **do**
  $(j, l_1) \leftarrow argmin_{il}\{Q_i\}$;
  remove (j, l_1) *from* Q_{il};
  if $j == t$ **then**
    stop while;
  for *each* k *in neighbors of* j **do**
    $l_2 \leftarrow$ *label index of arc* (j, k);
    $C_{kl_2} \leftarrow D_{jl_1} + c_{jk} + q_{ijk}$;
    if $C_{kl_2} < D_{jl_2}$ **then**
      $D_{kl_2} \leftarrow C_{kl_2}$;
      $P_{kl_2} \leftarrow j$;
      $Q_{kl_2} \leftarrow C_{kl_2} + B_k$;

3.2.3 Algorithm Adaptations

On both proposed algorithms, aqDijkstra and aqA^*, adjacent quadratic costs q were assumed to be pre-calculated and allocated on some existing data structure. Although having the quadratic costs allocated on memory improves the search speed of the algorithm, it can become a problem as instances increase in size. When dealing with real life graph instances, for example, graphs can easily achieve dimensions of million nodes, tens of millions of arcs and hundreds of millions of quadratic arcs. Thus, storing this information in memory becomes quite challenging.

Another improvement to aqDijkstra and aqA^* is suggested, which embeds the calculation of adjacent quadratic costs on the algorithms. It is assumed that there exists a function $\Gamma : \tilde{A} \rightarrow R^+$, able to calculate quadratic costs based on adjacent arc indices (i, j, k). The adapted version of aqDijkstra algorithm is

presented on Algorithm 4 and the modification can be used to embed quadratic calculation on aqA^*. For the sake of simplicity, the adapted algorithms will be named Adapted aqDijkstra and Adapted aqA^*.

Algorithm 4: Adapted Adjacent Quadratic A^*

input : G, c, s, t
output: D, P

$B \leftarrow Dijkstra(G, t)$;
initialize D_{ij};
initialize Q_{il};
$D_{sl} \leftarrow 0$;
for *each node* v *in* V **do**
    for *each label* l *in* $L(v)$ **do**
        if $v \neq s$ **then**
            $D_{vl} \leftarrow Inf$;
            $P_{vl} \leftarrow$ *origin node from label* l;
while $Q_{il} \neq \emptyset$ **do**
    $(j, l_1) \leftarrow argmin_{il}\{Q_i\}$;
    remove (j, l_1) *from* Q_{il};
    if $j = t$ **then**
        stop while;
    for *each* k *in neighbors of* j **do**
        $l_2 \leftarrow$ *label index of arc* (j, k);
        $q_{ijk} \leftarrow \Gamma(G, i, j, k)$;
        $C_{kl_2} \leftarrow D_{jl_1} + c_{jk} + q_{ijk}$;
        if $C_{kl_2} < D_{kl_2}$ **then**
            $D_{kl_2} \leftarrow C_{kl_2}$;
            $P_{kl_2} \leftarrow j$;
            $Q_{kl_2} \leftarrow C_{kl_2} + B_k$;

4
Computational Experiments

This section aims to evaluate the performance of the proposed algorithms, presenting results from a set of computational experiments. It starts by comparing the algorithms' speed and performing a benchmark analysis. Then, it explores random graph instances generated with two different methods and observes their effects on search behavior. On the third test, it is presented an analysis of algorithms' sensitivity to quadratic cost variation. The last experiment concerns a stress test, where instances are increased to dimensions up to half-billion quadratic arcs.

4.1
Software and Machine Settings

The Julia programming language were used throughout this work. The packages JuMP, Ipopt, SimpleWeightedGraphs were chosen to handle graph data structures and perform general optimization. Julia is widely known for its optimization capabilities, specially through the use a domain-specific modeling package called JuMP. In addition, Ipopt solver for convex and non-convex problems, has an interface with the language through the Ipopt package, which was used to solve the QP problem. Furthermore, SimpleWeightedGraphs provides good data structures to handle graph objects and great methods to solve SSP and ASQPP. It should be highlighted that the implementation of the aqDijkstra was made on top of Dijkstra implementation provided by the package.

The R programming language was also used in this work, mainly for GIS purposes. The packages rgdal and rgeos are widely known in the R community and were used for manipulating the geographical data that was further converted into the graphs used for computational experiments and application study.

All scripts used and implemented in this thesis are available for further verification, validation, and additions to the implemented model. To easy distribution and verification, a Julia package called AQSPP was created and is made available for use. Finally, computational calculations were made on an Intel Core i7-10700K 4.8GHz with 64 GB of RAM machine.

4.2 Graph Instances

The graphs used in the following experiments are essentially of two natures: grid and random instances. Grid graphs are deeply related to routing applications since nodes are evenly spaced and connected to their immediate neighbours. The grid-like format facilitates associating the graph to physical topologies, as well as other attributes related to the topology. On the other way, random graphs are important to study the search process of algorithms and evaluate performance under non-predictable conditions. By experimenting on random graph structures, with distinct levels of degree and sampled connections, one can better understand general performance of an algorithm. Both graph types are better detailed in the following sections.

4.2.1 Grid Graphs

On grid graphs, each node is associated with a real location on the globe and the arc costs represent the distances between locations. The graphs are built out of regularly spaced grid of elevation points, which we define on matrix 4-1 as the matrix Y with N rows and M columns. Each element of the matrix is associated with a numeric value and spatial coordinates.

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & \ddots & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix} \tag{4-1}$$

To create an graph instance, we convert the matrix Y into a weighted graph $G = (V, A)$, where each node i is equivalent to element y_{nm}, $|V|$ is the product of matrix dimensions $N \times M$, and $|A|$ can vary according to a graph composition rule. This rule is a function of how many neighbors n each node sees. Figure 4.1 presents the effect of choosing different number of neighbors on $|A|$.

The intuition behind node neighborhood comes from direction and length of movement. Common choices for n are 2, 4 and 8. Choosing $n = 2$ or $n = 4$ represent unitary movements two directions (east and south) or four directions (west/east and north/south), respectively. For $n = 8$, diagonal movements are contemplated.

We denote $d(y_{nm}, y_{op})$ as a function that calculates the distance cost between points y_{nm} and y_{op}. Thus, arc costs are defined as $\{c_{ij} = d(y_{nm}, y_{op}) \mid i = (n, m) \ \ and \ \ j = (o, p)\}$, where (n, m) and (o, p) are neighbors according to a given graph composition rule. For this work, we define

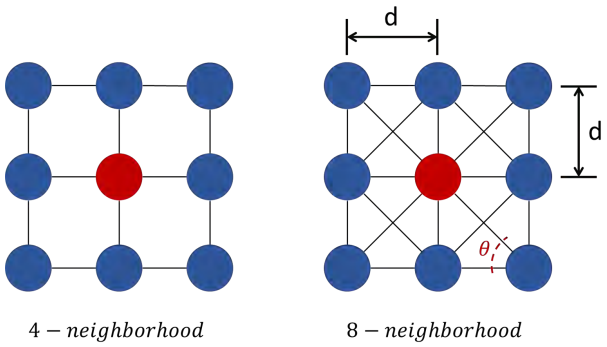


Figure 4.1: Demonstration of node neighborhood for $n = 4$ and $n = 8$

$n = 8$ so that diagonal movements are allowed. As we are working with spatial information, diagonal costs must be adjusted by a factor of $\sqrt{2}$.

4.2.2 Random Graphs

Random graphs can be defined as graphs associated to probability distributions. In other words, they are graphs described by a probability distribution or generated by a random process. The random graph sets build for experiments come from two different methods: Erdös-Rényi (Erdös and Rényi, 1959) and Configuration Model (Newman, 2010).

The Erdös-Rényi method defines a graph $G(n, p)$, where the arcs are added between pairs of nodes according to a given probability $p \in [0, 1]$. The greater p is, greater is the number of arcs, thus the more connected is G.

Moreover, the Configuration Model generates a random graph $G(n, k)$ from given degree sequence $k = \{k_i\}_{i=1}^{n}$. A degree defines the number of connections a given node has, therefore, as k increases, so does the number of arcs in G. To build $G(n, k)$, the method randomly samples k_i nodes from G and connects them to node i.

4.3 Benchmark Analysis

The computational experiments start by evaluating total search-time for all proposed algorithms and other approaches existing in technical literature. One of these approaches is an algorithm proposed by Rostami et al. (2015) and reviewed by Hu and Sotirov (2018), that solves AQSPP for directed acyclic graphs. This algorithm re-writes the original graph according to the linearization of adjacent quadratic costs. Thus, the optimal path can be found on the linearized graph through traditional SPP algorithms. For the sake of notation, this approach will be referred as the Linearization Algorithm for the

following sections. Another benchmark approach is formulating the AQSPP as a quadratic problem, such as described in Chapter 2 with a standard solver to perform the optimization.

Figure 4.2 presents the solution time for all studied algorithms. The figure is divided into two main plots due to differences in speed magnitude. On the first plot, results are displayed for Linearization Algorithm, aqDijkstra and Adapted aqDijkstra algorithms. On the second, results for $aqA*$ and it's adapted version. The y-axis refer to total search-time and the x-axis to the number of quadratic arcs on each instance.

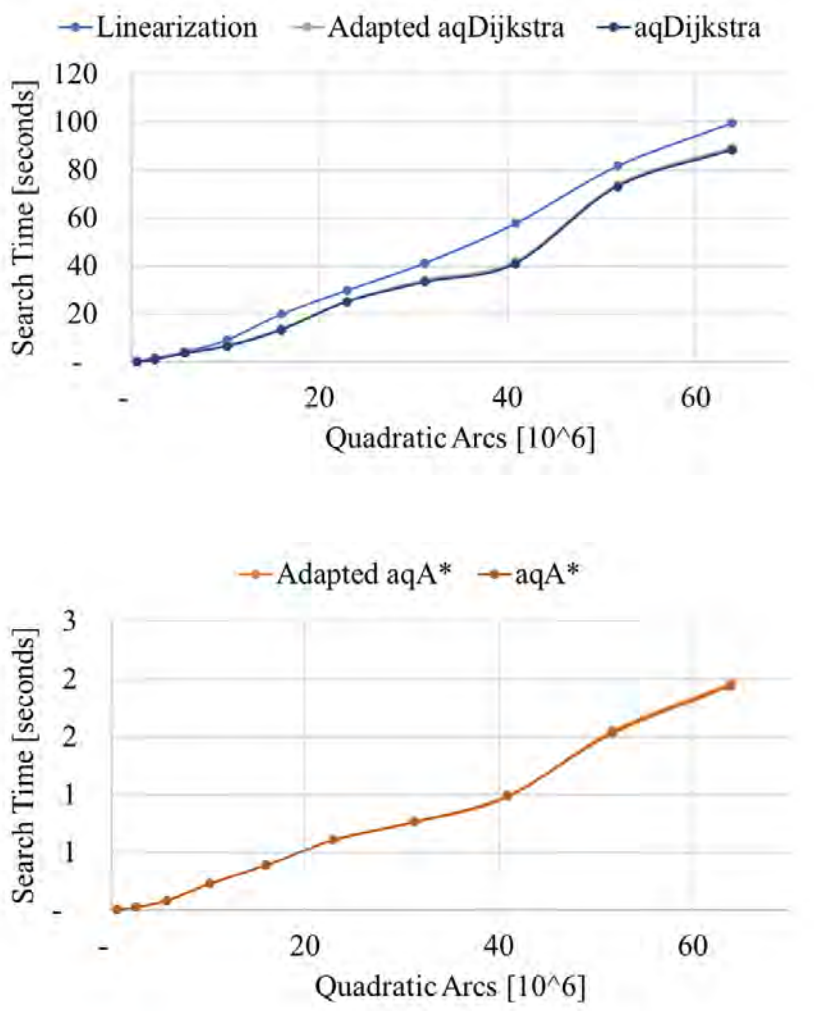


Figure 4.2: Benchmark Analysis: Search-Time Comparison

In addition, Figure 4.3 presents relative improvements in speed between the algorithms. This figure is also divided into two main plots due to differences in magnitude. The first plot displays the speed ratio between the Linearization approach and aqDijsktra. The other plot displays relative improvement of aqA^* in relation to Linearization's approach and aqDijkstra algorithm. The y-axis

refer to speed ratio and the x-axis to the number of quadratic arcs on each instance.

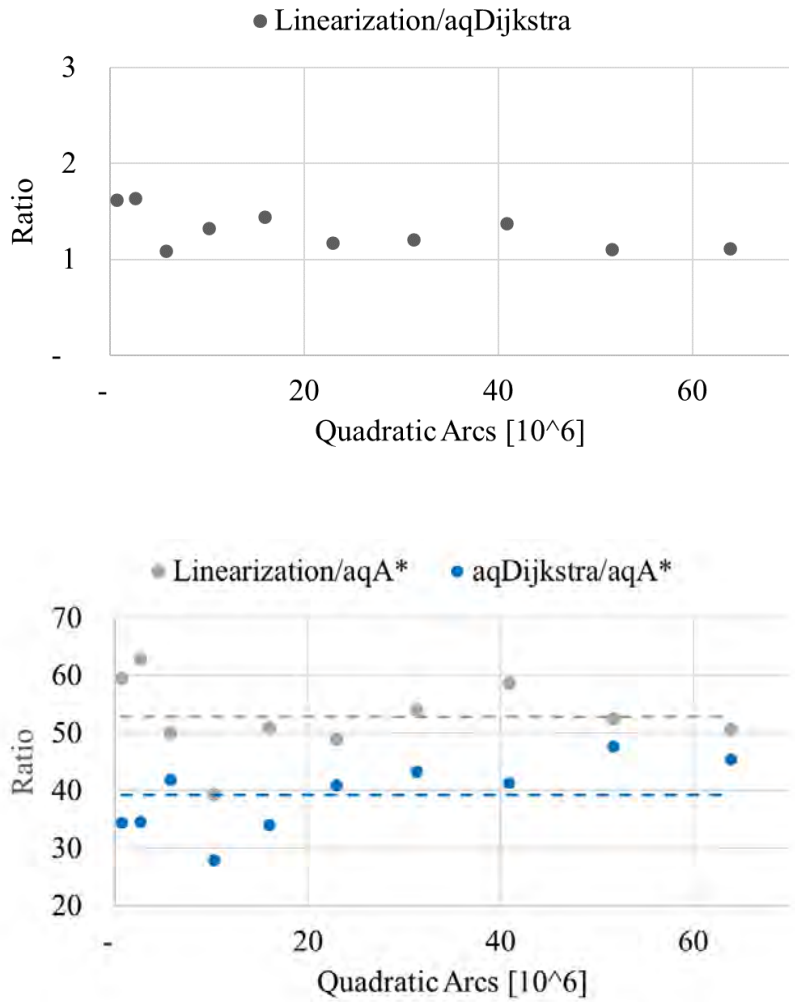


Figure 4.3: Benchmark Analysis: Relative Improvements

Numerical results seen on Figure 4.2 present the aqA^* as the fastest algorithm among all AQSPP algorithms tested. The speed ratios on Figure 4.3 measurements indicate that aqA^* algorithm is 39 times faster than aqDijkstra and 53 times faster than the Linearization method on average. Also, it seems the ratios fluctuate around this average value, which could suggest a constant ration. A possible explanation for the out-performance can be related to a similarity of paths obtained from SPP and AQSPP algorithms, resulting in very precise cost-to-go estimators, thus speeding up aqA^*.

Considering the benchmark with the Linearization approach and Quadratic Programming, both were outperformed by the algorithms proposed by this work. Results displayed on Figure 4.3 show aqDijkstra's improvements in speed from 10% to 50% in comparison to Linearization's approach. Further

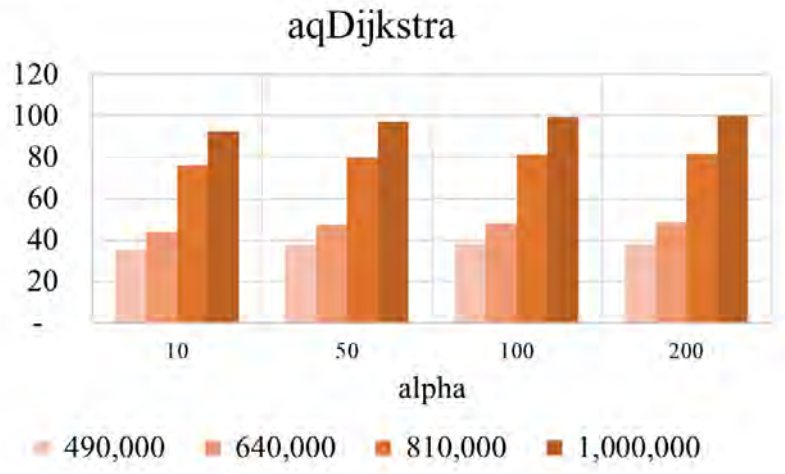
analysis revealed that the time spent to linearize the graph is very significant and becomes a downside to the Linearization algorithm, especially as instances grow. Furthermore, the results from the Quadratic Programming approach were not displayed due to solver's inability to solve even the smallest instance tested for this experiment.

Moreover, it can be noticed the similarity between original and adapted versions, for both aqDijkstra and aqA^*. Figure 4.2 shows that, although calculating quadratic costs on-the-run lowers search speed, the impact is insignificant. Since the adapted versions have excellent benefits on memory allocation and allow solving larger instances, their use is highly recommended for applications where adaptation is viable resource.

4.4 Cost Variation Analysis

The second experiment measures the sensitivity of the algorithms to the magnitude of quadratic costs. A variable $\alpha \in \{10, 50, 100, 200\}$ was set to scale the quadratic costs, so that arc interaction cost would be $\alpha\ q_{ijk}$. For each value of α, four graph instances ranging from 490×10^3 to 1×10^6 nodes were evaluated.

Figure 4.4 presents total search-time measurements for all approaches, for each instance and α values. The y-axis defines the range of total search-time among the approaches. Alpha values are defined on x-axis. The instances are classified by number of nodes and labeled at the bottom. Results are displayed on individual plots, one for each method studied.



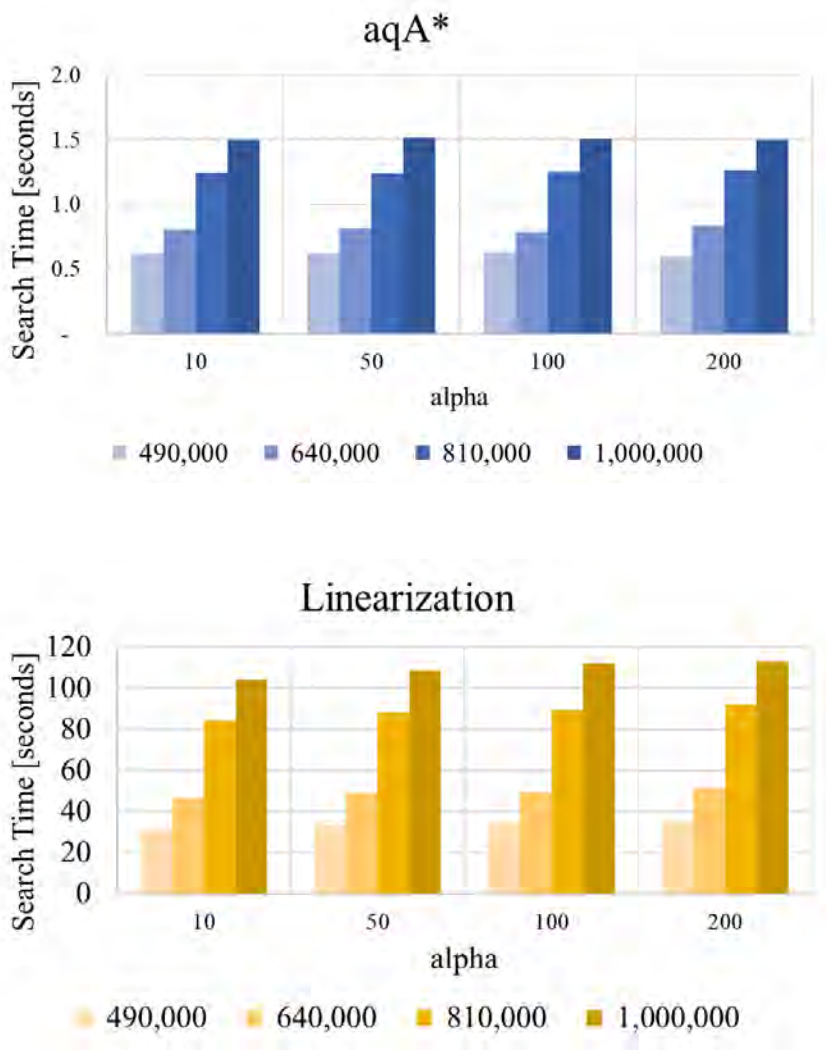


Figure 4.4: Quadratic Cost Variation Analysis: Search-Time Comparison

Results presented on Figure 4.4 suggest that aqA^*, aqDijkstra and Linearization Algorithm are not significantly influenced by an increase of quadratic costs. No apparent trends or unconventional behavior are seen on the plots, confirming the insensitivity towards cost variation.

This result was particularly interesting for aqA^*, since it has a linear cost-to-go estimators that would be expected to lose relevance as quadratic costs increase. However, the α factors chosen for this experiment were not sufficient to turn linear costs-to-go into bad estimators. In other words, even with quadratic costs scaled-up by a factor of 200, the linear estimators still provided great "guidance" for the search process. This behavior highlights the robustness of the algorithm.

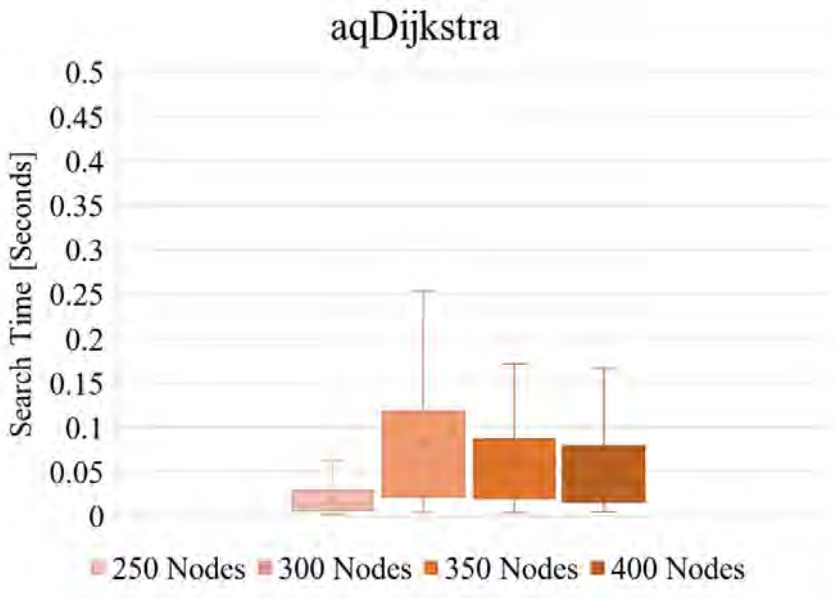
4.5 Random Graph Instances Analysis

On this third experiment, the goal is to study the algorithms' performance considering randomly generated graph instances. The main idea is to identify with this experiment how random changes on graph topologies and arcs costs impact the search process. The methods used to build the instances are known as Erdös-Rényi and Configuration Model and have been widely used in the technical literature.

For the Erdös-Rényi method, it was defined $p = 0.8$ to obtain very connected instances and vary the number of nodes $n \in \{100, 150, 200, 250, 300, 350, 400\}$. For each configuration (n, p), 100 random instances were generated. Costs for arcs and quadratic arcs were also randomly chosen from uniform distribution, so that $(q_{ijk}, c_{ij}) \sim Unif(\xi) \mid \xi \in [0, 1]$.

In addition, on the Configuration Model, the node degrees were fixed to 8 and vary the number of nodes $n \in \{150 \times 10^3, 200 \times 10^3, 250 \times 10^3, 300 \times 10^3\}$. Similarly, 100 random instances were generated with c_{ij} and q_{ijk} draw from an $Unif(\xi) \mid \xi \in [0, 1]$.

Figure 4.5 present the compiled results from this experiment. The plots on the top display the search times for each algorithm and instance size tested, considering the 100 different topologies generated with Erdös-Rényi method. The same results are displayed for the instances generated using the Configuration Model on Figure 4.6.



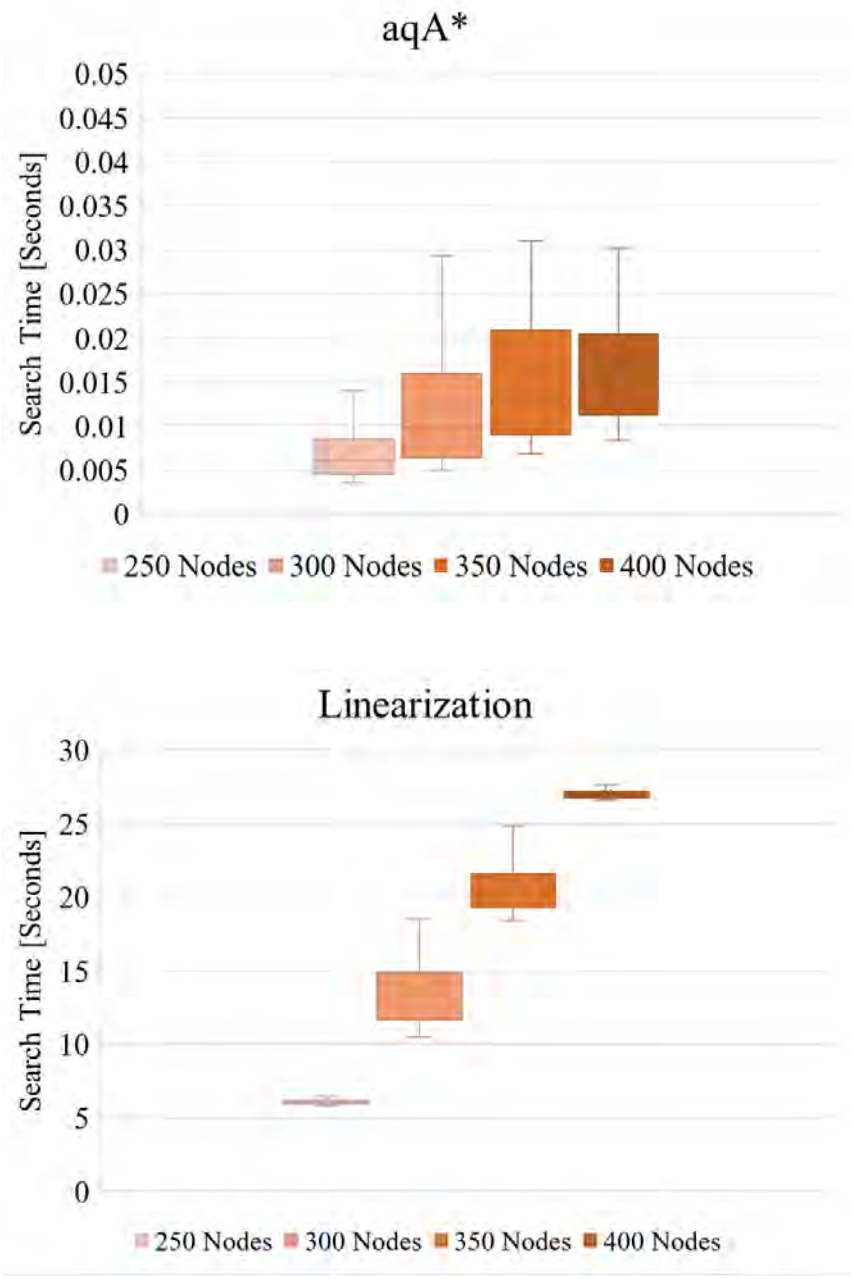
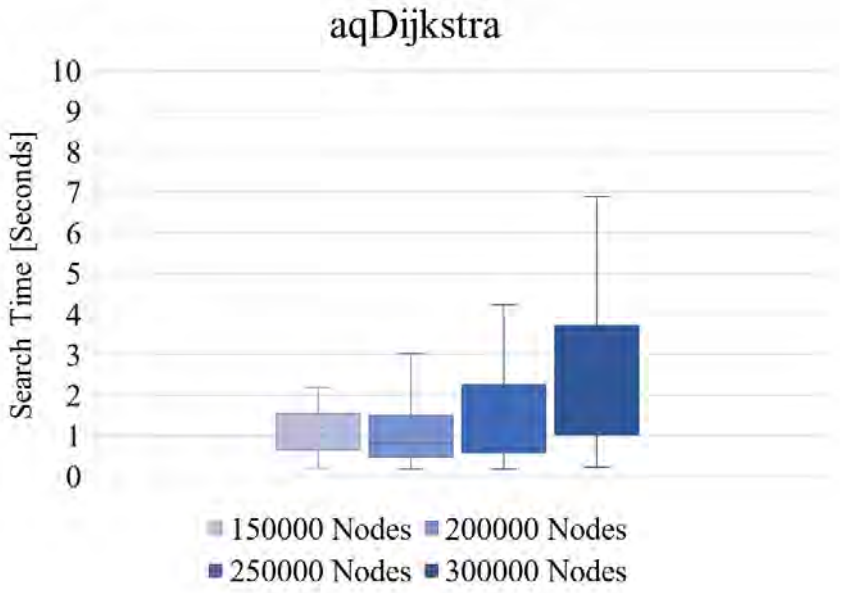
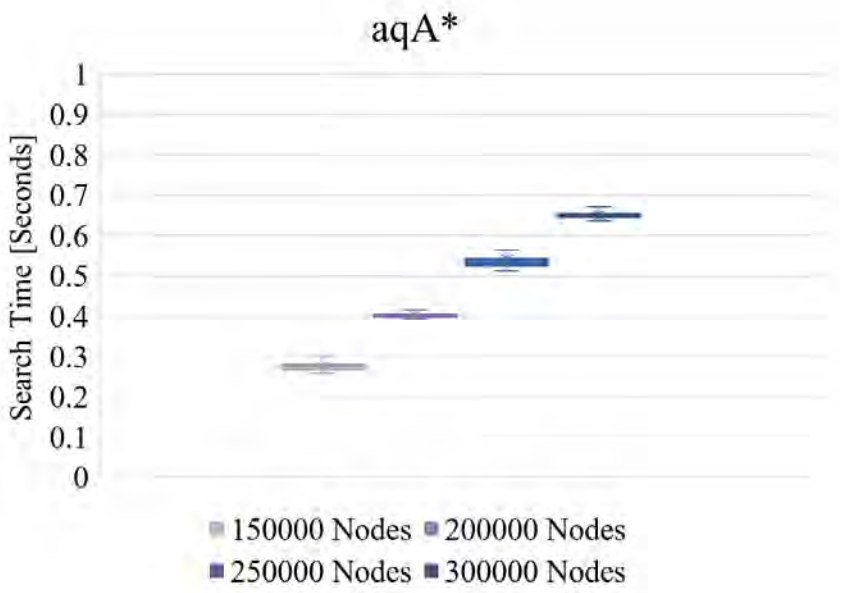


Figure 4.5: Analysis of Erdös-Rényi Instances: Search-time spread

The analysis starts by looking at the results from instances generated with the Erdös-Rényi method. Figure 4.5 presents noticeable variation in search-time for each one of the studied algorithms. Results for aqA^* and aqDijkstra showed very low and similar search-times through the instances. This can be explained by the high connectivity of Erdös-Rényi instances. Since p was set 0.8, there is a high probability the source node is connected to the target node, or that the shortest path contains only few nodes. As consequence, the number of nodes needed to be visited becomes small, shortening the search process.

Furthermore, magnitude of results highlight differences of the Linearization technique, in comparison to aqA^* and aqDijkstra. This is primarily caused by the time spent to build the linearized graph. Since the number of quadratic arcs is large, performing the linearization is very time-consuming and becomes a drawback to this approach.





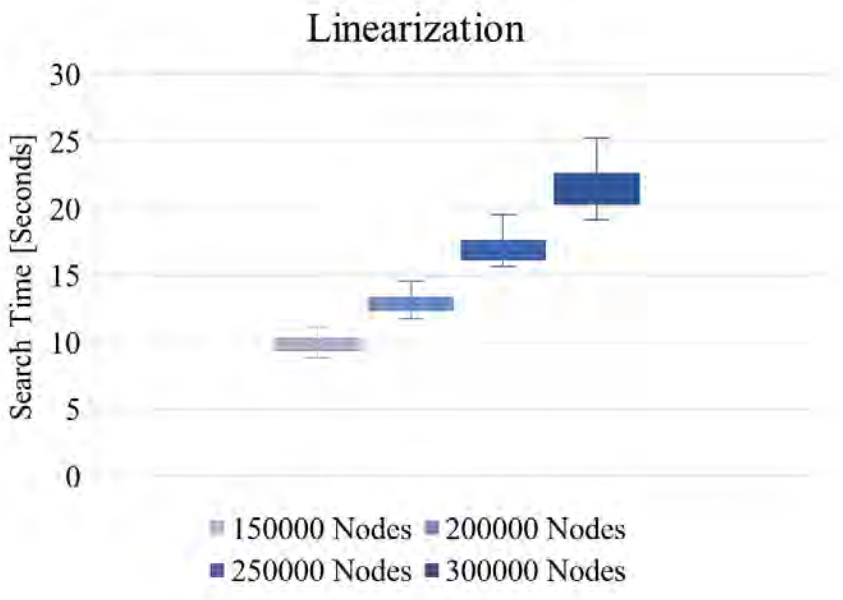


Figure 4.6: Analysis of Configuration Model Instances: Search-time spread

Results from the Configuration Model, displayed in Figure 4.6, suggest that the instance size and the observed variance are positively correlated, which is better noticed in aqDijkstra and Linearization's approach. It can also be recognized a positive tendency of growth in the search time that follows instances sizes, as expected.

Furthermore, data presented indicate that the aqA^* exhibits much less variance thus resulting in more consistent and predictable search time for instances generated with the Configuration Model. This becomes more evident in contrast to aqDijkstra results, which demonstrate greater variance and search time magnitude. It can be argued that these results reaffirm the role of the backward step on providing good cost-to-go estimates and consistently speeding up the search process.

4.6 Stress Analysis

A final experiment was designed to stress out search capabilities on very large instances, ranging from 64×10^6 to 576×10^6 quadratic arcs. Since real applications may require path search on immense instances, studying algorithms' performances over increasingly larger graph instances can provide meaningful insights.

Figure 4.7 displays the results from these experiments. The first presents the search-time measurements for the Linearization Algorithm, aqDijkstra and it's adapted version. The y-axis defines search-time and the x-axis defines

the number of quadratic arcs on each instance. The second follows the same structure, but compile results for aqA^* and it's adapted version.

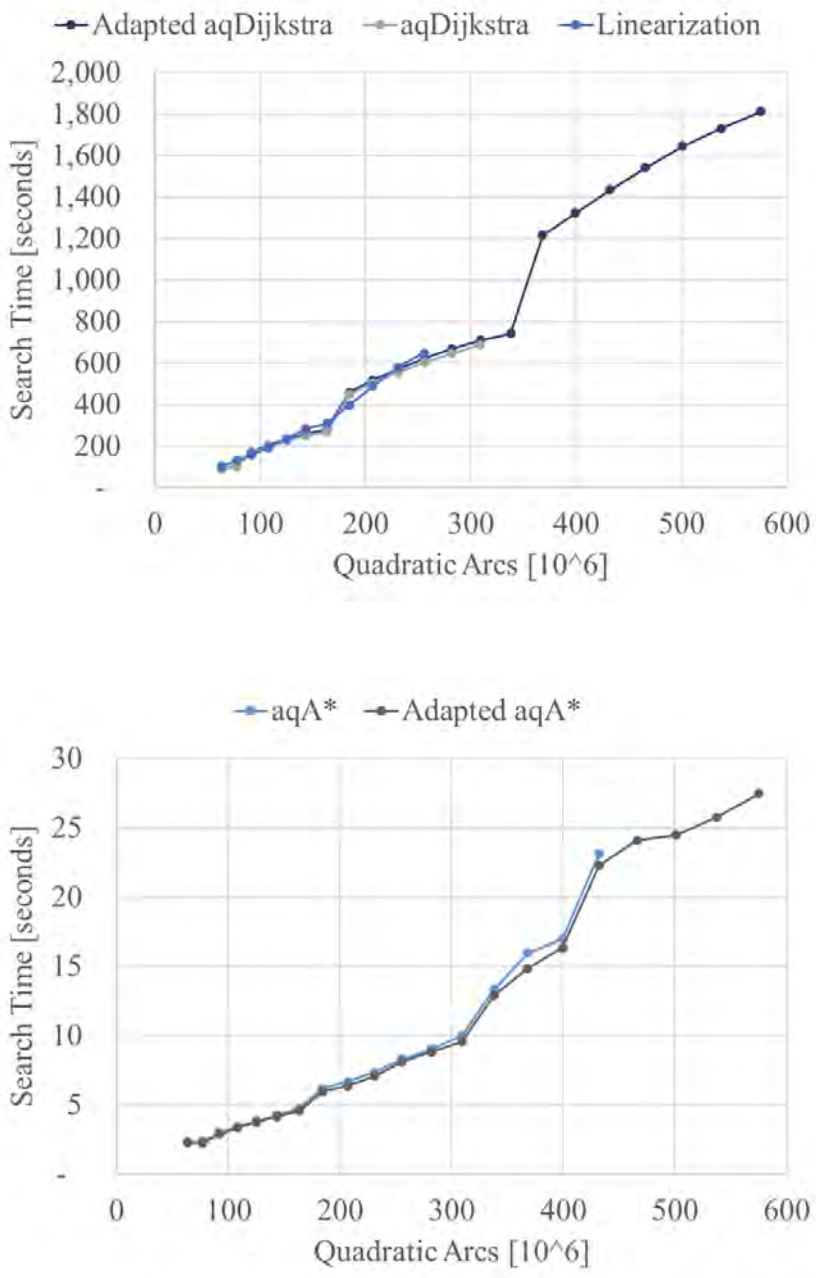


Figure 4.7: Stress Analysis: Search-Time Comparison

Results are presented on Figure 4.7 and show that none of the non-adapted algorithms were able provide solutions for all instances, due to memory problems. Since the methods assume a pre-existing structure that maps adjacent arcs into quadratic costs, storing this data can become problematic as the sizes of instances increase. Furthermore, aqDijkstra and aqA^* have internal data structures to store relevant data throughout the search process, which also

requires memory space. The Linearization's approach had even higher demand for memory allocation since there was a need to store the linearized graph.

Overall, aqA^* achieved better results, being able to solve a graph instance with 432×10^6 quadratic arcs. Linearization Algorithm and aqDijkstra Algorithm had somewhat similar extensions, solving instances up to 309×10^6 and 255×10^6 quadratic nodes, respectively.

In contrast, the adapted versions of aqDijkstra and aqA^* were able to solve all instances tested, as it is presented on Figure 4.7. Since quadratic calculations are made throughout the search, these algorithms do not require excessive memory space, allowing them to properly handle increasingly larger instances. As expected, adapted aqA^* remained the faster approach, solving the AQSPP in 27.48 seconds on a graph with 9×10^6 nodes, 36×10^6 arcs and 576×10^6 quadratic arcs.

5
Application

This chapter provides two applications of transmission line routing to highlight the use of the methodology developed in this dissertation. The first application is an illustrative example designed to ease understanding towards the decision-process of SPP and AQSPP algorithms and point out their differences. Then, a real application is presented for the routing of a future transmission line in Brazil, connecting substation Poções III and substation Medeiros Neto II. This last case explores the benefits of AQSPP formulation on real studies and discusses how they could improve planning practices.

5.1
Illustrative Example

The first example describes the search process of ASQPP and SPP algorithms on a simple graph. The instance comes from a 3 X 3 matrix, resulting into 9 nodes. Considering the number of neighbors as 8, the final graph would have 40 undirected arcs. Quadratic costs are given and count 200 adjacent arcs. The list of linear and quadratic costs be found on appendix.

5.1.1
Spatial Data Setup

According to the methodology described on Chapter 3, the initial step is the Spatial Data Setup. On this example, the final cost map is considered given, thus assuming the steps described on Sections 3.1, 3.1.1 and 3.1.2 were previously performed. The final cost map is displayed on Figure 5.1. This map is then converted into the weighted graph shown in Figure 5.2, following the conversion process described on Section 3.1.3.

2	3	2
2	5	1
3	4	2

Figure 5.1: Cost Map

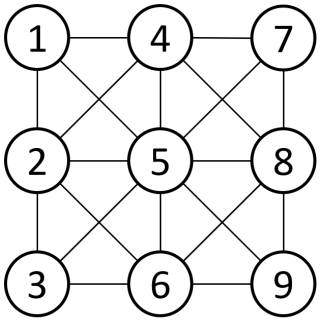


Figure 5.2: Conversion to Graph

5.1.2 Route Optimization

In following, the route is optimized through the search-path algorithm chosen. Figures 5.3 and 5.4 present the optimal paths resulted from SPP and AQSPP approaches. The algorithm used for SPP was the Dijkstra Algorithm, which is the traditional technique used on these types of application. In contrast, the aqDijkstra Algorithm was used to solve the AQSPP.

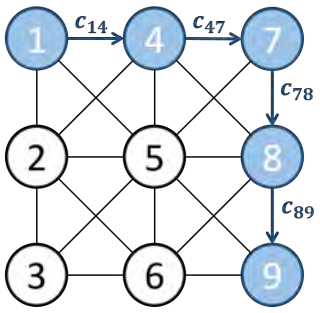


Figure 5.3: SPP Path

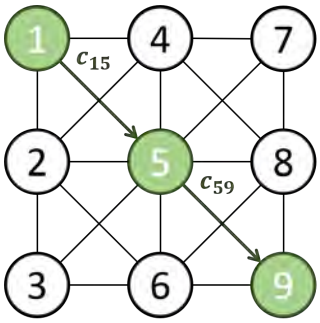


Figure 5.4: AQSPP Path

The difference is solution is evident on Figures 5.3 and 5.4. Since the SPP approach only considers arc costs, it found the path which minimizes the sum of arc costs. As expected, Table 5.1 presents lower total linear costs for SPP solution. However, when curvature penalties are considered, the path changes to the AQSPP solution assuming a straight line, connecting node 1 and 9 through node 5. This solution does not result in any deflection angles throughout the route, thus not adding quadratic costs. As also expected, Table 5.1 presents higher quadratic costs for the SPP path.

Table 5.1: Linear and Quadratic cost from SPP and AQSPP approaches

Algorithm	Linear Costs	Quadratic Costs	Total Costs
Dijkstra [SPP]	$c_{14}+c_{47}+c_{78}+c_{89}=5.64$	$q_{147}+q_{478}+q_{789}=4.24$	9.88
aqDijkstra [AQSPP]	$c_{15}+c_{59}=8.94$	$q_{159}=0.00$	8.94

5.2 Real Case

The methodology and novel routing approach also have been applied to a real case of transmission line planning. The chosen line was a single-circuit between substations Poções III and Medeiros Neto II operating at a 500kV Voltage level. According to the official expansion plan, the TL is expected to start its operation on 2026. The line was planned for expansion to improve transfer capacity between Northeast and Southeast Regions.

Most of Brazil's load comes from the Southeast, which has the biggest cities and the most advanced industrial parks. This fact alone explains the need for transmission expansion towards the region. Moreover, the Northeast region has been retaining a great portion of generation expansion for the next years. Its massive potential for renewable power, combined with significative drops on solar and wind technology prices, attracted investors and big generation projects.

The substation Poções III is named after the city Poções located in Bahia, one of the 9 states of the Northeast region. Medeiros Neto II is also named after a city, Medeiros Neto, which borders the Southeast state of Minas Gerais. Studies from the Brazilian Planning Agency (EPE) describes the candidate line as:

- Conductor: 6 x TERN (795 MCM)
- Line Reactance: 0.0140 [Ω/km]
- Line Resistence: 0.1917 [Ω/km]
- Flow Capacity: 1129 [MVA]
- Emergency Flow Capacity: 1267 [MVA]
- Instalation Cost: 1.234 [mi R$/km]

These parameters and cost estimation comes from different planning studies performed by the agency. They are divided into 5 reports, which covers technical, economic and environmental detailed analysis. Two reports

are especially important for this application, the first (Technical Report R1, 2020) and the third (Technical Report R3, 2020). An initial evaluation of TL candidates is done in R1 through the definition of power corridors, which is a 10 to 20-kilometer area surrounding a preliminary route. Furthermore, R3 performs a detailed environmental-social analysis within the selected corridor and proposes a final route.

The methodology proposed on this dissertation covers both reports and improves the representation of technical constraints, thus providing more realistic routes. Improving route calculation can bring benefits for the planner and for the entrepreneur responsible for the future project. Realistic routes allow better cost estimations since they mirror actual construction practices, such as avoiding high deflection angles. On the builder perspective, better candidate routes reduce uncertainty towards the actual building of the line, thus reducing total spends. In addition, planning agencies can better assess the real cost of transmission expansion, increasing energy market efficiency through proper transmission tariffs.

In this context, the Brazilian System Planner has an import role deciding which new power transmission lines must be built and where it will be located. Every year, EPE publishes an official document describing the plan (PDE) for expanding Brazil's electrical system (EPE, 2020). The PDE indicates the prospects for a 10-years expansion horizon, discussing requirements for future generation and transmission infrastructure. According to the 2029 plan (PDE29) more than 55,785 kilometers of transmission lines will be built within the next 10 years, summing up to almost 73 billion reais of expected investment in transmission infrastructure.

The following sections provide further an overview of the studied area according to methodology guidelines and a detailed routing analysis from SPP and AQSPP solutions.

5.2.1 Spatial Data Setup

As described on Chapter 3, the methodology starts with a selection of relevant spatial data to the area studied. Initially, a tool called EPE Web Map was used to cross spatial layers of transmission infrastructure and environmental-social constraints. The tool was developed by EPE and works as an interactive map. Due to its public nature, EPE Web Map is a free resource. Furthermore, existing reports from the planner were also used to filter data. Table 5.2 compiles the list of public data collected and assumptions for additional costs. The values set to ∞ refer to spatial areas in which the

route shouldn't trespass. Section 3.1.2 details how they are modeled into the cost maps.

Table 5.2: List of considered spatial constraints and associated costs assumptions

Spatial Constraint	Source	Year	Additional Cost [mi$/km]
Topological Map	Embrapa	2005	1.23
Urban Areas	IBGE	2018	9.44
Water Bodies	IBGE	2015	12.55
Indigenous Land	INCRA	2015	∞
Conversation Units	INCRA	2017	∞
Rural Areas	INCRA	2017	4.92
Federal Roads	Embrapa	2018	0.00
Railways	Embrapa	2018	0.00
Transmission Lines	EPE	2021	0.00
Archaeological Sites	IPHAN	2020	∞

Figure 5.5 filters the available spatial data for the studied area, which contains the two substations. One can first notice existing transmission infrastructure, environmentally protected areas, traditional communities, water bodies, railways, federal roads, urban areas. According to their intersection with the final studied area, these attributes are converted into additional construction costs.

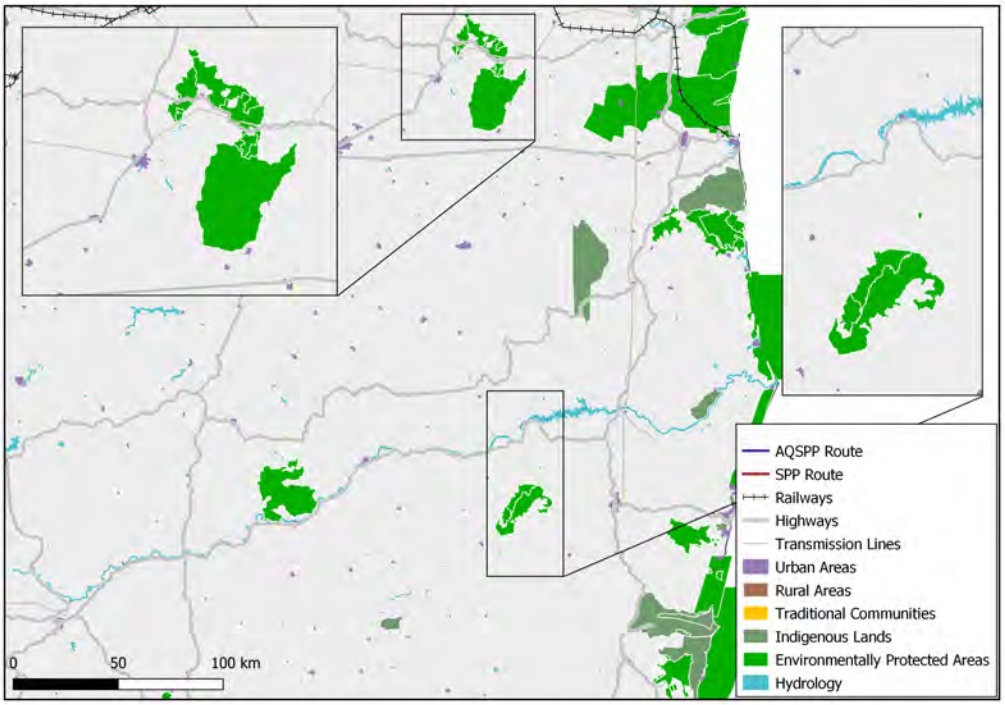


Figure 5.5: Relevant spatial areas

Some specific areas should be cited due to their relevance and possible interference on final route. The areas are the following: city of Poções, city of Medeiros Neto, Environmentally Protected Area of Serra do Ouro, National Park of Alto Cariri and Archaeological Sites close to Itamaraju city. The first two cities are hold the substations being studied. Although the methodology can identify higher construction costs within urban areas, the actual building is very complex and, many times, require further modification on the proposed route. Serra do Ouro is a 506 km^2 protected area due to the environmental richness of the region, such as portions of Atlantic Forests and dense hydrology. National Park of Alto Cariri is also legally protect for environmental aspect, thus laying out obstacles for construction.

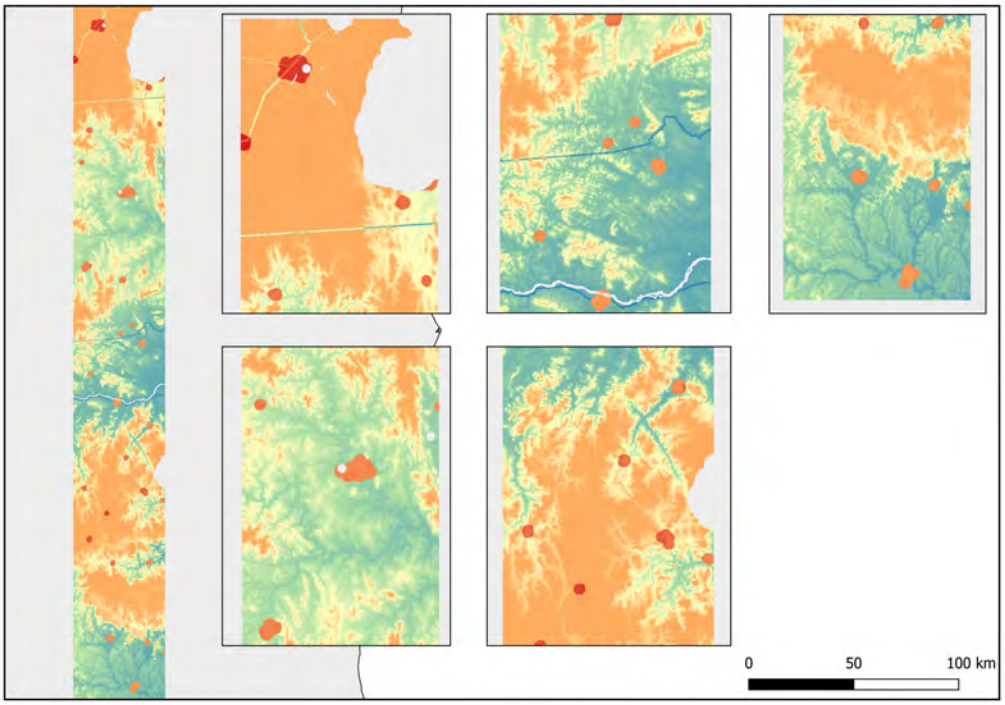


Figure 5.6: Final construction cost map

The tower costs described on Table 5.3 are relative to different steel structures recommended for 500kV single-circuit lines. Each tower has specific structures that allow it support a range of deflection angles. Naturally, higher the angle is, the stronger the structure must be, increasing total cost. The costs of Table 5.3 are estimated by the Midcontinent Independent System Operator (MISO) and compiled on a Transmission Cost Estimation Guide (MISO MTEP19, 2019). MISO is an independent, not-for-profit organization that delivers power across 15 U.S. states and the Canadian province of Manitoba.

Table 5.3: Tower Relative Costs

Structure	Angle Range	Total Cost [k\$]	Relative Cost [%]
Tangent	0^o to 2^o	176.6	1.00
Strain	2^o to 30^o	385.4	2.18
Dead-end	$> 45^o$	551.6	3.12

For this application, the relative costs are considered a proxy of curvature penalties, according to angle range support. In other words, quadratic costs to be considered by ASQPP are a function of deflection angles found throughout the route. Angles are calculated using the equations presented on Section

3.1.4 of Chapter 3. In addition, function $\Gamma(\theta_{ijk})$ it is defined as $\Gamma(\theta_{ijk}) = \frac{c_{ij}+c_{ij}}{2} \times w(\theta_{ijk})$ where $w(\theta_{ijk})$ is the relative cost of angle θ_{ijk}.

5.2.2 Route Optimization

The optimal routes are displayed on Figure 5.7, along with the cost map and spatial constraints' layers. The graph instance built for this problem had 15,497,220 nodes, 61,952,021 undirected arcs and 991,232,336 quadratic arcs. Once this graph is built, SPP and AQSPP approaches can be used to find the optimal route. For SPP, Dijkstra's algorithm was used. Considering results from Chapter 4, specifically from the stress test on Section 4.6, the adapted version of aqA^* algorithm presented itself as the best approach for this application. The quadratic costs are defined according to the cost of Table 5.3 and assigned dynamically within the algorithm. The aqA^* algorithm was executed on the same machine used for computation experiment (section 4.1) with total search-time of 4152 seconds.

Figure 5.7 displays the candidate routes on top of the full map. A first look point to routes noticeably different, as it could be expected due to the different formulations. It can be further noticed how the routes seem to converge on departure and arrival and diverge on the middle segment of the route. This initial evaluation point to a significant impact of AQSPP formulation on routing decisions.

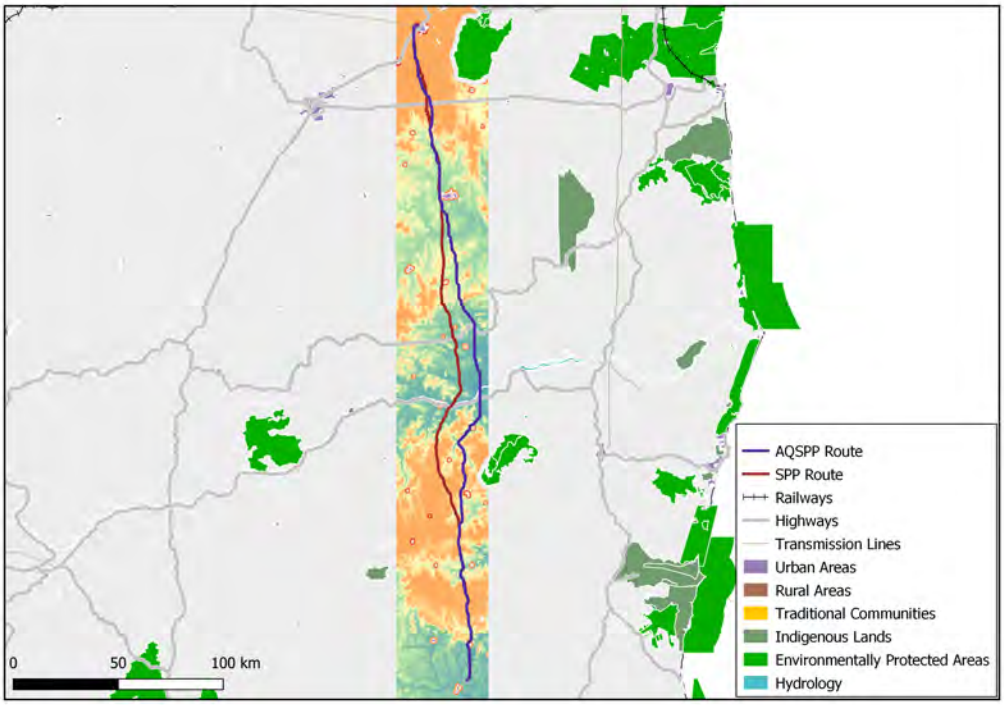


Figure 5.7: SPP and AQSPP Optimal Routes

Moreover, Table 5.4 measures curvature aversion of each approach based on direction shifts on the route. As described on 3.1.4, each direction shift results on a specific deflection angle, which is penalized accordingly on ASQPP. Results show almost six times less occurrences of 45º and a reduction of 90º inflections by an order three. Both inflection angles are direct related to the $n = 8$ criteria for node-neighborhood, as described on 3.1.3. Increasing n to $n = 16$ or $n = 32$ would allow accessing lower angles, such as 30º or 15º for example, allowing better representation of curvature penalties on AQSSP. Although desirable, as n grows, so does the graph instance, making it harder to store and optimize.

Table 5.4: Angle Deflection Overview

Algorithm	0º	45º	90º
Dijkstra [SPP]	8689 (82.7%)	1731 (16.5%)	74 (0.8%)
aqA^* [AQSPP]	10024 (96.6%)	326 (3.2%)	26 (0.2%)

Figures 5.8 and 5.9 zoom into departure substation Poções III and arrival substation Medeiros Neto II, respectively. The departure region contains

environmentally protected areas, highways, existing transmission infrastructure and an urban area representing the city of Poções. Since urban areas are expensive to build, a straight-line pattern is seen on an attempt to minimize cost. The overall intense reddish tonality of the cost map indicates high construction costs related to terrain slope, which pushes both routes to seek relatively close paths. The first 20 kilometers present similar tracing patterns, with significant divergence afterwards. Initial impacts of curvature penalties can be seen as AQSPP seeks longer straighter lines to avoid penalties. This behavior is also evident on routes' arrival at Medeiros Neto II. On Figure 5.9 only three inflections are seen on AQSPP route, in contrast to twelve on SPP route.

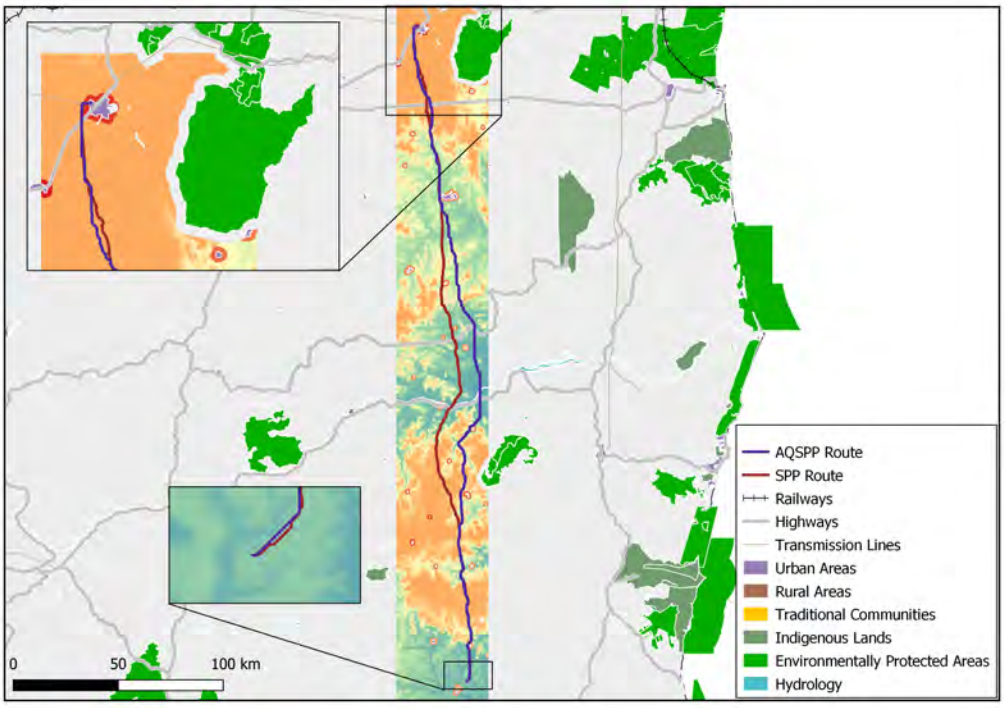


Figure 5.8: 2D view of route departure (Poções II) and arrival (Medeiros Neto III)

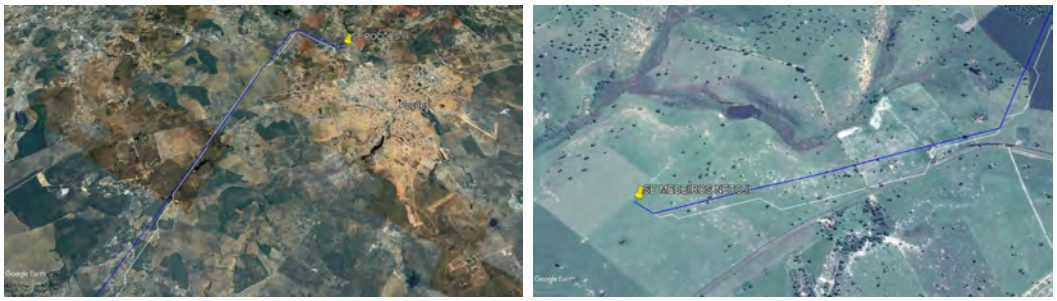


Figure 5.9: 3D view of route departure (Poções II) and arrival (Medeiros Neto III)

A major route detour can be observed on the middle section of the studied area, as Figure 5.10 illustrates. The disparity suggests that ASQPP forced an east-direction movement so it could benefit built large sections of straight lines at lower construction costs, thus avoiding curvature penalties. These lower costs are indicated by the bluish tonality of the cost map. Similarly, diagonal movements of ASQPP are noticed on the bottom section of the area, aiming to avail lower costs from a valley region. The route also contours an urban area, as expected.

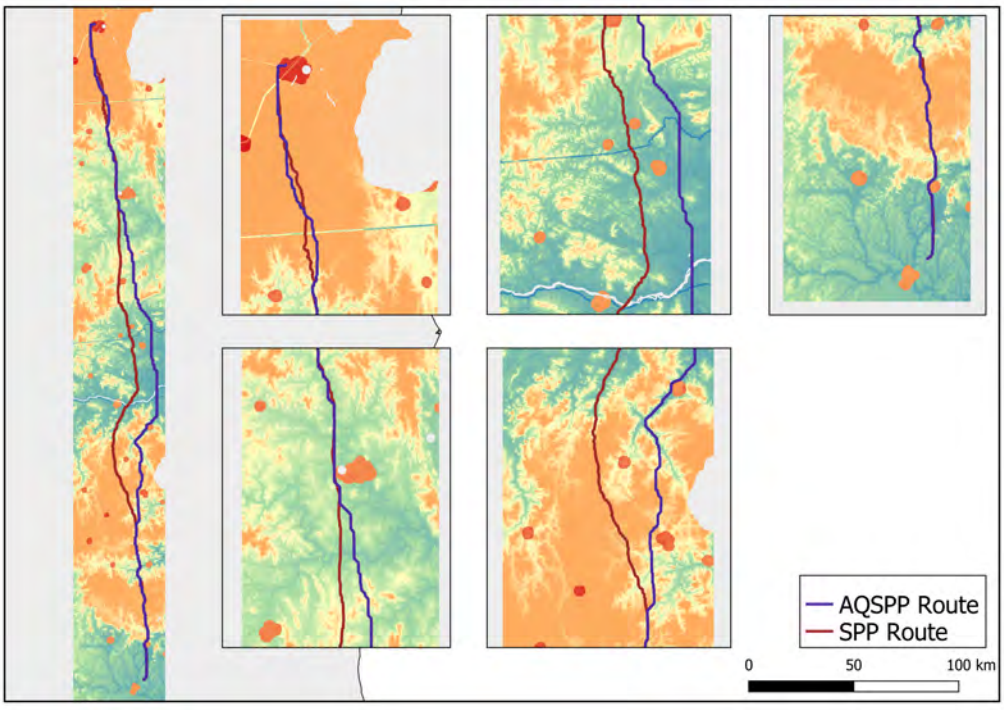


Figure 5.10: SPP and AQSPP route major divergence

Furthermore, one can explore the required contouring of Ibirajá city

for even greater contrast of AQSPP and SPP routing decisions. Figure 5.11 displays a satellite view of the city and routes, along with the two individual cost maps, one for each approach. Both routes follow the same guideline for countering, sharing spatially-close vertices, but are significantly discrepant. SPP attempt to minimize line length while avoiding high costs of urban areas, resulted in unrealistic vertex selection, thus provoking several inflection angles.

In major contrast, AQSPP route presented smoother and realistic patterns of routing, emulating real building decisions. The penalization of inflection angles allowed aqDijkstra to evaluate the cost-benefit of decreasing line length and increasing curvature costs. This event strengthens the argument for properly representing curvature penalties, as they played a key role in emulating actual building decisions.

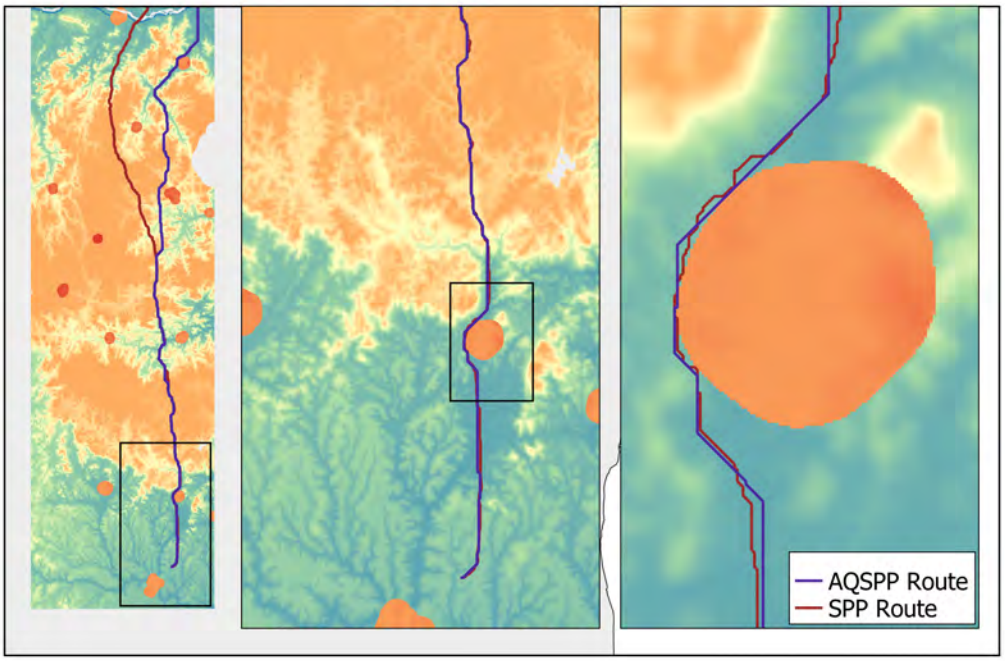


Figure 5.11: Contrast of AQSPP and SPP routes towards city contouring

Final results are presented on Table 5.5 as a summary of candidate's length, reference costs of construction and additional expenses (curvature penalties). The reference costs were previously defined as 1.234 kR$/km according to estimations presented on preliminary reports from the system planner. Additional costs represent the total amount of curvature penalties for the final route given by Dijkstra (SPP) and aqA^* (AQSPP) algorithms. Notice that quadratic costs are not natural outcomes of SPP, thus being calculated after route optimization.

The result shows SPP route as the shortest candidate and AQSPP route as the candidate with least curvature penalties. According to a traditional perspective of TL routing, the SPP alternative is preferable. The line was 7.5 miR$ cheaper and 6.1 kilometers shorter than ASQPP route. However, once quadratic costs are considered, additional expanses for the SPP route increase, shifting preference towards AQSPP alternative. The total curvature penalties for AQSPP route were 35.7 miR$, being three times lower than penalties for SPP. This would represent a 8.6% increase in total construction costs.

Table 5.5: Linear and Quadratic cost from SPP and AQSPP approaches

Algorithm	Distance [km]	Reference Cost [$]	Additional Costs [$]
Dijkstra [SPP]	330.8	408.2	111.3
aqDijkstra [AQSPP]	336.9	415.7	35.7

6
Conclusion and future work

This dissertation provides theoretical description of Adjacent Quadratic Shortest Problems in the context of transmission line routing and proposes an extension of Dijkstra's ($aqDijkstra$) algorithm to solve AQSPP for graphs with no improving cycle. Two improvements for the main approach are also presented in this work: an adjacent quadratic version of A^* with backward cost-to-go estimation and an embedded calculation of adjacent quadratic costs. The work also discusses in detail how AQSPP formulation can improve transmission line routing methodologies and generate more realistic routes.

On the theoretical perspective, empirical evidence is presented and suggests that $aqDijkstra$ and aqA^* can solve AQSPP in polynomial time. As the graph instances increase in size, the search time ratio between Linearization approach and $aqDijkstra$ remains constant. This is also valid for aqA^*.

It is also shown through computational experiments that $aqDijkstra$ outperformed the algorithm proposed by Rostami, being 10% to 50% faster. This is also valid for the proposed algorithm aqA^*, which was the fastest approach among the ones studied, outperforming aqDijktra's search speed by an order of 40 on the benchmark analysis.

Further test results suggest that the a proposed algorithms also had great performance under randomly generated instances, pointing to their robustness. In addition, a major drawback of the Linearization Technique was identified for highly connected graphs. This problem is related to the time it spent performing the linearization.

Moreover, the algorithm's seemed to be indifferent to adjacent quadratic cost variation, which was already expected for $aqDijkstra$ and an interesting result for aqA^*. Results suggests that even for high quadratic-linear cost ratio, the backward search is still deeply relevant to aqA^*.

Applications presented in this work allowed visual evaluation of the impact of quadratic costs on routing. As expected, routes given by aqA^* were significantly different from Dijkstra's, reducing the number of high deflection angles on the route. Results show almost six times less occurrences of 45^o and a reduction of 90^o inflections by an order three. Since neighbor criteria of $n = 8$ limits the angles to 0^o, 45^o and 90^o angles, future works can explore how

changing the criteria to $n = 16$ or $n = 32$ can improve the final result.

The content of this dissertation provides an active contribution to the scientific community in terms of graph theory and spatial routing research. Results from the computational experiment enriches AQSPP research and motivates further works. Moreover, modeling tower-angle constraints as quadratic cost tackles a real technical problem and brings innovation to transmission line routing methodologies.

Future work should be done to consolidate even more this area of study. Possible works can explore the impact of different quadratic cost functions, evaluate upsides and downsides for increasing node-neighborhood criteria, benchmark ASQPP routes with other known routing methodologies for different applications, such as gas pipeline routing, and other relevant subjects. The main recommended themes for research are:

- Increase node-neighborhood criteria to $n = 16$ or $n = 32$ to access smaller angles
- Explore AQSPP formulation for different applications, such as gas-pipeline routing
- Study the behavior of tower siting algorithms on SPP and AQSPP alternatives
- Evaluate the real cost of AQSPP routes through specialized costing software

Bibliography

AKGÜN, V.; ERKUT, E. ; BATTA, R.. **On finding dissimilar paths**. European Journal of Operations Research, 121:232–246, 2000.

BELLMAN, R. E.. **On a routing problem**. The Quarterly of Applied Mathematics, 16:87–90, 1958.

BUCHHEIM, C.; TRAVERSI, E.. **Quadratic 0-1 optimization using separable underestimators.** Technical Report, Optimization Online, 2015.

CAPRARA, A.. **Constrained 0-1 quadratic programming: Basic approaches and extensions.** European Journal of Operational Research, 187(3):1494–1503, 2008.

CHOOBINEH, F.; BURGMAN, T.. **Transmission line route selection: An application of k-shortest paths and goal programming**. IEEE Trans. Power Syst, 11, 1984.

DEMIRCAN, S.; AYDIN, M. ; DURDURAN, S. S.. **Finding optimum route of electrical energy transmission line using multi-criteria with qlearning**. Expert Syst Appl, 38:3477–3482, 2011.

DIJKSTRA, E.. **A note on two problems in connexion with graphs**. Numer. Math., 1:269–271, 1959.

EPE. **Plano decenal de expansão de energia 2029**, 2020.

ERDÖS, P.; RÉNYI, A.. **On random graphs**. Publicationes Mathematicae, 6:290–297, 1959.

EROGLU, H.; AYDIN, M.. **Optimization of electrical power transmission lines' routing using ahp, fuzzy ahp, and gis**. Turk J Elec Eng and Comp Sci, 23:1418–1430, 2015.

EROGLU, H.; AYDIN, M.. **Solving power transmission line routing problem using improved genetic and artificial bee colony algorithms**. Springer-Verlag GmbH Germany, part of Springer Nature, 2018.

FANG, S.; ROY, S. ; KRAMER, K.. **Transmission Structures Structural Engineering Handbook**. CRC Press LLC, 1999.

FLOYD, R. W.. **Algorithm 97: Shortest path**. Communications of the ACM, 5(6):345, 1962.

GONÇALVES, V.; BAPTISTA, E.; CAMPOS, G. M. ; HOFFMANN, L.. **Transmission line routing optimization using rapid random trees**. Electric Power Systems Research, 194, 2021.

HART, P.; NILSSON, N. ; RAPHAEL, B.. **Formal basis for the heuristic determination of minimum cost paths**. Systems Science and Cybernetics, 4(2):100–107, 1968.

HOBBS, B. F.; XU, Q.; HO, J.; DONOHOO, P.; KASINA, S.; OUYANG, J.; PARK, S. W.; ETO, J. ; SATYAL, V.. **Adaptive transmission planning**. IEEE Power and Energy Magazine, p. 30–40, 2016.

HU, H.; SOTIROV, R.. **Special cases of the quadratic shortest path problem**. Journal of Combinatorial Optimization, 2017.

HU, H.; SOTIROV, R.. **A polynomial time algorithm for the linearization problem of the qspp and its applications**. arXiv, 1802(02426v1), 2018.

KIESSLING, F.; NEFZGER, P.; NOLASCO, J. F. ; KAINTZYK, U.. **Overhead power lines, planning design construction**. Springer, 2004.

LUMBRERAS, S.; RAMOS, A. ; BANEZ-CHICHARRO, F.. **Optimal transmission network expansion planning in real-sized power systems with high renewable penetration**. Electric Power Systems Research, 149:76–88, 2017.

MISO. **Transmission cost estimation guide**, 2019.

MADKOUR, A.; AREF, W. G.; REHMAN, F. U.; RAHMAN, M. A. ; BASALAMAH, S.. **A survey of shortest-path algorithms**. 2017.

MARTINELLI, R.; CONTARDO, C.. **Exact and heuristic algorithms for capacitated vehicle routing problems with quadratic costs structure.** INFORMS Journal on Computing, 27(4):658–676, 2015.

MONTEIRO, C.; RAMIREZ-ROSADO, I.; V. MIRANDA, P. Z.-S.; GARCIA-GARRIDO, E. ; FERNANDEZ-JIMENEZ, L.. **Gis spatial analysis applied to electric line routing optimization**. IEEE Transactions on Power Delivery, 20:934–942, 2005.

MURAKAMI, K.; KIM, H. S.. **Comparative study on restoration schemes of survivable atm networks**. Proceedings IEEE, 1:345–352, 1997.

NEWMAN, M.. **Networks: An introduction**. Oxford University Press, 2010.

PIVETEAU, N.. **A novel approach to the routing problem of overhead transmission lines**. Master's thesis, University of Zurich, Alte Landstrasse 158, 8800 Thalwil - Switzerland, 2017.

ROSTAMI, B.; MALUCELLI, F.; FREY, D. ; BUCHHEIM., C.. **On the quadratic shortest path problem**. E. Bampis (ed.) Experimental Algorithms, Lecture Notes in Computer Science, Springer International Publishing, 9125:379–390, 2015.

ROSTAMI, B.; CHASSEIN, A.; HOPF, M.; FREY, D.; BUCHHEIM, C.; MALUCELLI, F. ; GOERIGK., M.. **The quadratic shortest path problem: complexity, approximability, and solution methods**. Optimization Online, 2018.

SANTOS, A.; DE LIMA, R.; PEREIRA, C.; OSIS, R.; MEDEIROS, G.; DE QUEIROZ, A.; FLAUZINO, B.; CARDOSO, A.; JUNIOR, L.; DOS SANTOS, R. ; JUNIOR, E.. **Optimizing routing and tower spotting of electricity transmission lines: An integration of geographical data and engineering aspects into decision-making**. Electric Power Systems Research, 176, 2019.

SHANDIZ, S. G.; DOLUWEERA, G.; ROSEHART, W. D.; BEHJAT, L. ; BERGERSON, J. A.. **Investigation of different methods to generate Power Transmission Line routes.** Eletric Power Systems Research, 165:110–119, 2018.

EPE. **Análise técnico-econômica e socioambiental de alternativas: Relatório r1**, 2020.

NEOENERGIA. **Relatório de definição da diretriz de traçado e análise socioambiental: Relatório r3**, 2020.

THOMAS, B.; CALOGIURI, T. ; HEWITT, M.. **An exact bidirectional a^* approach for solving resource-constrained shortest path problems**. Networks, 73(2):187–205, 2019.

VELASQUEZ, C.; WATTS, D.; RUDNICK, H. ; BUSTOS, C.. **A framework for transmission expansion planning**. IEEE Power and Energy Magazine, p. 20–29, 2016.